Composite materials and lightweight construction parts are increasingly replacing classic materials in practical engineering applications. However, it is often difficult to simulate their mechanical behavior due to the complex structure, where a full FEM discretization would lead to immense calculation times and memory requirements. While homogenization methods have been established to determine effective properties of composite microstructures, most of these approaches are unable to accurately reflect nonlinear system behavior. In this work a coupled multiscale approach is presented specifically for sandwich panels with comb-like core structures. A numerical homogenization is carried out in each iteration and load step of a nonlinear simulation, allowing to describe local effects of nonlinearity such as face sheet buckling or plastic flow. Several numerical simulations using different core geometries are carried out. The numerical results agree very well to full scale reference solutions, while drastically reducing the size of the equation system. Moreover, the method is well suited for parallelization and runs effectively on modern hardware topologies.

ISBN 978-3-935868-40-2

Band 40

A nonlinear shell model for comb-like sandwich panels

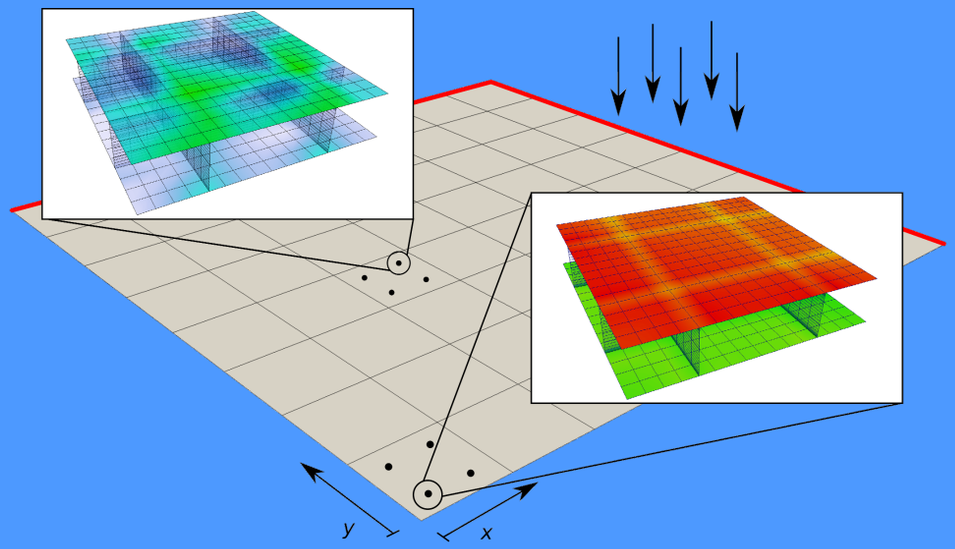
D. Heller

MECHANIK

FORSCHUNGSBERICHT

A nonlinear multiscale finite element model for comb-like sandwich panels

Dominik Heller



TECHNISCHE UNIVERSITÄT DARMSTADT

Band 40

A nonlinear multiscale finite element model for comb-like sandwich panels

Vom Fachbereich Bau- und Umweltingenieurwissenschaften
der Technischen Universität Darmstadt

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)

genehmigte

Dissertation

von

Dipl.-Math. Dominik Heller
aus Münster (Westfalen)

Erstreferent:	Prof. Dr.-Ing. habil. F. Gruttmann
Korreferent:	Prof. Dr.-Ing. habil. W. Wagner
Tag der Einreichung:	29.09.2015
Tag der mündlichen Prüfung:	27.11.2015

Darmstadt 2016
D 17

Heller, Dominik

A nonlinear multiscale finite element model for comb-like sandwich panels

Forschungsberichte des Instituts für Mechanik der Technischen Universität Darmstadt

Band 40

Herausgeber der Reihe:
Studienbereich Mechanik, Technische Universität Darmstadt

Verfasser:
Dominik Heller

Verlag:
Studienbereich Mechanik, Technische Universität Darmstadt
Franziska-Braun-Straße 7, 64287 Darmstadt

Druckerzeugung:
Lasertype GmbH, Darmstadt

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Creative Commons Lizenz (CC BY-NC-ND 3.0 DE), 2016.
Freies Vervielfältigen und Weiterverbreiten – Namensnennung –
Nicht-kommerziell – Keine Bearbeitung

ISBN 978-3-935868-40-2

Acknowledgement

This thesis is a result of my research and work at the Chair of Solid Mechanics at Technische Universität Darmstadt. Its completion would not have been possible without the help of certain important people. First and foremost, I wish to express my gratitude to my supervisor Professor Dr.-Ing. Friedrich Gruttmann, who provided me with the opportunity of conducting research in a very interesting field, always took the time for answering my questions, and offered me much help and guidance during my work. He supported me in combining the research on finite element methods with my interests in scientific computing and programming on many occasions over the last years, for which I am particularly thankful. I am lucky having had such good working conditions for my research. I also want to thank Professor Dr.-Ing. Werner Wagner, who readily accepted to co-supervise this thesis, for his interest in my work and for the fruitful collaboration in working with and improving FEAP. Moreover, I extend my gratitude to my current and former colleagues, especially Grischa Maaß and Simon Klarmann, who provided for a friendly work environment, productive discussions, and pleasant after work activities. In particular, I want to thank Mr. Jan Kratochvíl, who provided me with many valuable suggestions and has been a good friend during the mutual time in Darmstadt and ever since. I am very thankful to my family for their constant support and to my girlfriend Julia, who always knew how to ease me up when tension took over. Last, but not least, I want to thank Mrs. Angelika Priessnigg, who was always happy to help with all the not so minor details.

Darmstadt, September 2015 Dominik Heller

Meinen Eltern und Geschwistern.

Contents

Nomenclature v

Kurzfassung 1

Abstract 2

1 Introduction 5
- 1.1 Motivation . . . 5
- 1.2 Overview . . . 7
- 1.3 State of research . . . 9
- 1.4 Outline . . . 11

2 Basic principles of continuum mechanics 15
- 2.1 Reference configuration and current configuration . . . 15
- 2.2 Deformation and strains . . . 17
  - 2.2.1 Deformation gradient . . . 17
  - 2.2.2 Displacement gradient . . . 19
  - 2.2.3 Polar decomposition . . . 19
  - 2.2.4 Strain tensors . . . 19
- 2.3 Stress . . . 20
  - 2.3.1 Traction vector . . . 20
  - 2.3.2 Stress tensors . . . 21
- 2.4 Conservation laws . . . 22
  - 2.4.1 Conservation of mass . . . 22
  - 2.4.2 Conservation of linear and angular momentum . . . 23
  - 2.4.3 Conservation of energy . . . 24
  - 2.4.4 Entropy inequality . . . 25
- 2.5 Constitutive laws . . . 25
  - 2.5.1 Helmholtz free energy . . . 26
  - 2.5.2 J_2-plasticity for small strains . . . 28
- 2.6 Variational principles and linearization . . . 30
  - 2.6.1 Variational principle . . . 30

2.6.2 Linearization . . . 33
2.6.3 Mixed field formulations . . . 34

3 Nonlinear shell theory **37**
3.1 Shell kinematics . . . 38
3.1.1 Green-Lagrange strains and shell strains . . . 41
3.2 Stress and stress resultants . . . 43
3.3 Equilibrium conditions . . . 44
3.4 Mixed field formulation . . . 46

4 Finite element formulation **49**
4.1 Approximation of geometry . . . 49
4.2 Interpolation and variation of Green-Lagrangean strains . . . 51
4.3 Second variation of the functional . . . 55
4.4 Interpolation of stress resultants and shell strains . . . 57
4.5 Linearized variational functional . . . 59

5 The coupled global-local shell model **61**
5.1 Extension to the local model . . . 61
5.2 Local assembly of the coupled system . . . 63
5.3 Boundary conditions for the RVE . . . 67
5.4 Simultaneous iteration . . . 69

6 Numerical modeling of sandwich structures **71**
6.1 Parameter terminology . . . 72
6.2 Unidirectionally stiffened sandwich panel . . . 72
6.2.1 Validation of RVE material matrix . . . 72
6.2.2 Axially stiffened sandwich beam . . . 78
6.2.3 Geometrically nonlinear deformation behavior . . . 83
6.3 Bidirectionally stiffened sandwich cell . . . 85
6.3.1 Validation of RVE material matrix . . . 85
6.3.2 Linear deformation behavior . . . 88
6.3.3 Comparison of stress resultants . . . 91
6.3.4 Geometrically nonlinear deformation behavior . . . 94
6.3.5 Physically nonlinear deformation behavior . . . 97
6.4 Hexagonal honeycomb sandwich cell . . . 108
6.4.1 RVE material matrix . . . 111
6.4.2 Honeycomb sandwich plate . . . 114

7 Mesh generation **119**
7.1 Automated mesh generation for FEAP using Python scripts 120
7.2 Conversion of Abaqus meshes using inp2feap 121

8 Parallelization and effective data transfer in a finite element program **125**
8.1 Parallel computer architectures 126
8.1.1 Increase in computational power due to more processing cores . . . 127
8.1.2 Memory-bound applications 128
8.1.3 Memory topologies in parallel machines 129
8.1.4 Potential benefits and drawbacks of hardware accelerators 129
8.2 Parallel programming issues and pitfalls 130
8.2.1 Race conditions 130
8.2.2 Deadlocks 131
8.2.3 Determinism of floating point arithmetics 132
8.2.4 Address locality and inter-process communication 132
8.3 FEAP shared memory extension 133
8.4 Shared memory performance 135

9 Conclusion and future perspectives **139**

Bibliography **144**

Nomenclature

Abbreviations and names

0	Zero (can be vector- or tensor-valued)
Abaqus	Commercial FEM code [1]
API	Application programming interface
BVP	Boundary-value problem
CCS	Corrugated core sandwich
CFRP	Carbon-fiber-reinforced polymer
DOF	Degree of freedom
FE2, FE^2	Multi-scale finite element method
FEAP	Research FEM code [51]
FEM	Finite element method
GRP	Glass-reinforced plastic
JSON	JavaScript Object Notation, data format
Matlab	Mathematical computing environment and programming language [92]
MPI	Message Passing Interface, parallelization standard [62]
OpenMP	Shared memory parallelization API [73]
Python	Interpreter-based scripting language [79]
RVE	Representative volume element
SMP	Shared memory processing
SRAM	Static random-access memory
TDP	Thermal design power

Greek letters

α, β	Indexes 1, 2
γ_α	Shell shear strains
$\boldsymbol{\gamma}$	Euler-Almansi strain tensor
Γ	Boundary of a domain Ω

δ	Variational operator
δ_{ij}	Kronecker symbol
ε	Longitudinal strain (1d)
$\varepsilon_{\alpha\beta}$	Shell membrane strains
$\boldsymbol{\varepsilon}$	Linearized strain tensor (Ch. 2); vector of shell strains (Ch. 3–9)
$\boldsymbol{\varepsilon}^{el}$	Elastic part of strain tensor
$\boldsymbol{\varepsilon}_g$	Geometric shell strain vector
$\boldsymbol{\varepsilon}_p$	Physical shell strain vector (independent field variable)
$\boldsymbol{\varepsilon}^{pl}$	Plastic part of strain tensor
$\boldsymbol{\theta}$	Vector of unknown field quantities
$\kappa_{\alpha\beta}$	Shell curvatures
λ	Load factor
ν	Poisson ratio
ξ, η	Isoparametric coordinates
ξ^i	Convective coordinate directions
Π	Potential energy
Π_{ext}	External potential energy
Π_{int}	Internal potential energy
ρ	Mass density in current configuration
ρ_0	Mass density in reference configuration
$\rho_{\alpha\beta}$	Second-order shell curvatures
σ	Normal stress (1d)
$\boldsymbol{\sigma}$	Linearized stress tensor (Ch. 2); vector of shell stress resultants (Ch. 3–9)
$\boldsymbol{\sigma}_i$	Vector of stress resultants in integration point i
$\boldsymbol{\sigma}_u$	Independent stress resultant vector
$\boldsymbol{\tau}$	Kirchhoff stress tensor
$\boldsymbol{\phi}$	Position vector in current configuration of a shell
$\boldsymbol{\Phi}$	Position vector in reference configuration of a shell
φ_t	Mapping from reference to current configuration
φ_t^{-1}	Mapping from current to reference configuration
$\hat{\varphi}$	Mapping from reference to current configuration
$\hat{\varphi}^{-1}$	Mapping from current to reference configuration
Ψ	Helmholtz free energy
ω	Length of pseudo-rotation vector
ω_i	Rotational parameters

$\boldsymbol{\omega}$	Rotation vector
Ω	Spatial domain
$\partial\Omega$	Boundary of a domain Ω
$\boldsymbol{\Omega}$	Skew-symmetric tensor

Roman letters

$\mathbf{a}_\alpha$	Tangent vectors to shell reference surface in current configuration
$\mathbf{A}$	Assembly matrix
$\mathbf{A}_\alpha$	Tangent vectors to shell reference surface in reference configuration
$\mathcal{B}$	Domain of a continuum body in Euclidean space
$\mathcal{B}_0$	Domain of a continuum body in the reference configuration
$\mathcal{B}_t$	Domain of a continuum body in the current configuration
$\partial\mathcal{B}$	Boundary of a continuum body in Euclidean space
$\partial\mathcal{B}_0$	Boundary of a continuum body in the reference configuration
$\partial\mathcal{B}_t$	Boundary of a continuum body in the current configuration
$\mathbf{b}_0$	Outer volume force or body force
$\mathbf{B}$	Left Cauchy-Green tensor
C^n	Space of n-times continuously differentiable functions
$\mathbb{C}$	Elasticity tensor
$\mathbf{C}$	Right Cauchy-Green tensor
$\mathrm{d}\mathbf{a}$	Area element in current configuration
$\mathrm{d}\mathbf{A}$	Area element in reference configuration
$\mathrm{d}v$	Infinitesimal volume in current configuration
$\mathrm{d}V$	Infinitesimal volume in reference configuration
$\mathrm{d}\mathbf{x}$	Line element in current configuration
$\mathrm{d}\mathbf{X}$	Line element in reference configuration
$\mathbf{d}$	Director vector in current configuration
D	Gâteaux directional derivative
$\mathbf{D}$	Director vector in reference config. (Ch. 3–4); material tangent (Ch. 5–9)
$\mathbf{D}_i$	Matrix of linearized stress resultants in integration point i
$\mathbf{D}^b$	Bending submatrix of linearized stress resultants
$\mathbf{D}^m$	Membrane submatrix of linearized stress resultants
$\mathbf{D}^{mb}$	Membrane/bending coupling submatrix of linearized stress resultants
$\mathbf{D}^s$	Shear submatrix of linearized stress resultants
$\boldsymbol{\mathcal{D}}$	Material tangent matrix, matrix of linearized stress resultants

e_0	Internal mechanical energy per unit volume
$\mathbf{e}_i$	Basis system in Euclidean space
E	Young's modulus
$\mathcal{E}$	Internal energy
$\mathbf{E}$	Green-Lagrange strain tensor
$\bar{\mathbf{E}}$	Green-Lagrange strain tensor in vector notation
$\hat{\mathbf{E}}$	Green-Lagrange strain tensor in shell reference surface
$\mathbf{f}_0$	Resultant of outer forces
$\mathbf{f}_e^L$	Element residual vector in a local scale boundary value problem
$\mathbf{F}_i^L$	Global residual vector in a local scale boundary value problem
$\mathbf{F}$	Deformation gradient
$\hat{\mathbf{F}}$	Deformation gradient in shell reference surface
$\mathbf{g}_i$	Covariant basis vectors in current configuration
$\mathbf{g}^i$	Contravariant basis vectors in current configuration
G	Shear modulus
$\mathbf{G}_i$	Covariant basis vectors in reference configuration
$\mathbf{G}^i$	Contravariant basis vectors in reference configuration
h	Height (length in z-direction)
h_x, h_y, h_z	Discretization parameters
h_x^R, h_y^R, h_z^R	RVE discretization parameters
h^+, h^-	Shell thickness coordinates
$\mathbf{H}$	Displacement gradient
i, j	Indexes 1, 2, 3
$\mathbf{I}$	Identity matrix or tensor
J	Determinant of deformation gradient
$\mathbf{J}$	Jacobian matrix
$\dot{\mathbf{J}}$	Time derivative of angular momentum
$\mathbf{k}^G$	Global scale element tangential stiffness matrix
$\mathbf{k}_e^L$	Element tangential stiffness matrix in a local scale boundary value problem
$\mathcal{K}$	Kinetic energy
$\mathbf{K}_T$	Global tangential stiffness matrix
$\mathbf{K}_i^L$	Global tangential stiffness matrix in a local scale boundary value problem
l_x	Length in x-direction
l_x^R	RVE length in x-direction
l_y	Length in y-direction

l_y^R	RVE length in y-direction
L	Linearization of functional
$\dot{\mathbf{L}}$	Time derivative of linear momentum
$m^{\alpha\beta}$	Shell bending moments
$\mathbf{m}_0$	Resultant outer moment
$\mathbf{m}^\alpha$	Director stress resultants
n_x	Number of cell walls with $x = \text{const}$
n_x^R	Number of cell walls in RVE with $x = \text{const}$
n_y	Number of cell walls with $y = \text{const}$
n_y^R	Number of cell walls in RVE with $y = \text{const}$
$n^{\alpha\beta}$	Shell membrane forces
$\mathbf{n}$	Unit normal vector in current configuration
$\mathbf{n}_0$	Unit normal vector in reference configuration
$\mathbf{n}^\alpha$	Stress resultants
neq	Number of equations
N	Size of linear equation system
N_I	Interpolation functions
$\mathbf{N}$	Unit normal vector
$\mathcal{O}$	Landau symbol, order of magnitude
$\bar{\mathbf{p}}$	Shell surface load
$\mathcal{P}_{ext}$	External mechanical power
$\mathcal{P}_{int}$	Internal mechanical stress power
$\mathbf{P}$	First Piola-Kirchhoff stress tensor
q_z	Line load or area load in z-direction
q^α	Shell shear forces
$\mathbf{q}_0$	Cauchy heat flow
$\mathcal{Q}$	Thermal power
$\tilde{\mathcal{Q}}$	Rate of ingoing entropy
$\mathbf{Q}$	Orthogonal tensor
r_0	Heat source
$\mathbf{r}$	Position vector to reference point
$\mathbf{R}$	Rotation tensor
$\mathbb{R}^+$	Set of positive real numbers
s_0	Local entropy per unit volume
$\mathbf{s}$	Stress deviator

$\mathcal{S}$	Entropy integrated over volume
$\mathbf{S}$	Second Piola-Kirchhoff stress tensor
$\hat{\mathbf{S}}$	Second Piola-Kirchhoff stress tensor in shell reference surface
t	Time; shell thickness
t_0	Initial time for reference configuration
t_{cw}	Shell thickness of cell walls
t_{fl}	Shell thickness of face layers
$\mathbf{t}$	Cauchy traction vector
$\mathbf{t}_0$	Outer boundary force vector
$\mathbf{t}_i$	Local base vectors
$\mathbf{T}$	Cauchy stress tensor
u_x, u_y, u_z	Displacement components in spatial directions x, y, z
$\mathbf{u}$	Displacement vector
$\delta\mathbf{u}$	Virtual displacement field
$\Delta\mathbf{u}$	Increment of displacement field
$\mathbf{U}$	Right stretch tensor
$\mathbf{v}$	Displacement/rotation vector
$\mathbf{v}_0$	Velocity vector of control volume
$\mathbf{V}$	Left stretch tensor (Ch. 2); vector of nodal displacements (Ch. 3–9)
$\mathbf{V}_i^L$	Global displacement vector in a local scale boundary value problem
w	Displacement in z-direction
W	Mechanical work
δW	Virtual work
δW_{ext}	External virtual work
δW_{int}	Internal virtual work
x, y, z	Spatial coordinates
$\mathbf{x}$	Position vector in current configuration
$\mathbf{X}$	Position vector in reference configuration
Y_0	Initial yield stress
$\mathbf{Z}$	Shifter tensor in reference configuration
$\bar{\mathbf{Z}}$	Shifter tensor in current configuration

Kurzfassung

Moderne Kompositwerkstoffe und Leichtbauteile ersetzen in der Praxis im Maschinenbau und Bauwesen zunehmend klassische Materialien. Durch die hohe Verbreitung entsteht ein Bedarf an Berechnungsmethoden, die das mechanische Verhalten dieser Bauteile akkurat beschreiben können, gleichzeitig aber eine zumutbare Rechenzeit gewährleisten. Eine klassische Finite-Element-Modellierung gerät im Hinblick auf Kompositwerkstoffe und Sandwichstrukturen an ihre Grenzen, da durch die vollständige Modellierung der Mikrostruktur enorm große FE-Modelle entstehen. Deren Anforderungen im Sinne von Rechenzeit und Speicherbedarf übersteigen die Rechenkapazität heutiger Computer in vielen Fällen bei weitem. Sogenannte Homogenisierungsverfahren ersetzen die vollständige Modellierung einer Mikrostruktur durch die Betrachtung eines homogenen Materials mit gemittelten Eigenschaften, die auf experimentelle, analytische oder numerische Art gewonnen werden können. Klassische Homogenisierungsmethoden scheitern jedoch in der Regel bei der Beschreibung nichtlinearen Systemverhaltens, da die effektiven Eigenschaften im Laufe einer Simulation als konstant angenommen werden.

In dieser Arbeit wird ein gekoppeltes global-lokales Verfahren speziell für Sandwichstrukturen mit wabenförmigem oder axial verstärktem Kern vorgestellt. Hierbei wird ein globales Modell, in dem die vollständige zu untersuchende Struktur mit herkömmlichen Schalenelementen diskretisiert wird, mit mehreren lokalen Modellen gekoppelt, welche die Sandwich-Mikrostruktur durch den gesamten Dickenverlauf nachbilden und ebenfalls mit Schalenelementen vernetzt sind. Die lokale Formulierung wird dabei als Materialgesetz für das globale Modell implementiert, so dass in jedem Integrationspunkt der globalen Struktur ein lokales Randwertproblem ausgewertet wird. Durch die erneute Auswertung der lokalen Modelle in jedem Iterationsschritt einer nichtlinearen Simulation können physikalisch und geometrisch nichtlineare Effekte abgebildet werden. Anhand einiger Beispiele wird gezeigt, dass es im Gegensatz zu klassischen Homogenisierungsmethoden beispielsweise möglich ist elasto-plastisches Materialverhalten oder prä- und postkritisches Beulverhalten zu beschreiben. Neben der Herleitung der theoretischen Grundlagen und der Vorstellung der gekoppelten Methode sowie von Beispielen wird zudem auf einige Details der Netzgenerierung und auf die Implementierung einer Datenschnittstelle mit hoher Bandbreite zwischen globalen und lokalen Modellen eingegangen.

Abstract

Modern composite materials and lightweight construction elements are increasingly replacing classic materials in practical applications of mechanical and civil engineering. Their high prevalence creates a demand for calculation methods which can accurately describe the mechanical behavior of a composite structure, while at the same time preserving moderate requirements in terms of numerical cost. Modeling the full microstructure of a composite by means of the classical finite element method quickly exceeds the capabilities of today's hardware. The resulting equation systems would be extremely large and unsuitable for solution due to their enormous calculation times and memory requirements. Homogenization methods have been developed as a remedy to this issue, in which the complex microstructure is replaced by a homogeneous material using averaged mechanical properties that are determined via experiments or by analytical or numerical investigation. However, classical homogenization methods usually fail as soon as nonlinear system behavior is introduced and the effective properties, which are presumed to be constant, begin to change during the course of a simulation.

In this work, a coupled global-local method will be presented specifically for sandwich panels with axially stiffened or honeycomb cores. Herein, a global model, in which the complete structure is discretized with standard shell elements, is coupled with multiple local models, describing the microstructure of the sandwich throughout the full thickness coordinate and using shell elements for discretization as well. The local formulation is implemented by means of a constitutive law for the global model, so that one local boundary value problem is evaluated in each integration point of the global structure. By reevaluating the local models in every iteration step in a nonlinear simulation, physical and geometrical nonlinearity can be described. For instance, it will be shown in numerical examples that elasto-plastic material behavior and pre- and postcritical buckling behavior can be described, contrary to most classical homogenization methods. Next to the derivation of theoretical fundamentals and the introduction of the coupled method as well as several numerical examples, additional chapters are detailing some issues concerning mesh generation and the implementation of a high-bandwidth data interface between global and local models.

Chapter 1

Introduction

1.1 Motivation

Materials used in practical engineering applications, such as mechanical engineering or civil engineering, are subject to ever-increasing requirements in terms of stiffness, strength and other parameters such as noise and heat insulation, vibration dampening, and more. Over the course of technological advances in manufacturing processes, more materials have become available and were adopted in practical use, increasingly replacing traditionally common materials such as steel, concrete, or glass.

One diverse family of materials is known as composite materials, classifying a combination of two or more individual component materials with generally very different physical properties. Exploiting the advantageous structural placement of its constituent materials, the composite material often features superior design properties such as a high specific stiffness. Common examples for modern composite materials include fiber-reinforced materials like carbon-fiber-reinforced polymer (CFRP) or glass-reinforced plastic (GRP), where generally multiple layers of a fiber-matrix material with different fiber orientations are stacked on top of each other as a laminate.

In contrast to that, this work will focus on so-called sandwich composites. These generally consist of two thin, but relatively stiff face layers (also called face sheets or skins) at the top and bottom surface of a composite structure, which are kept apart from each other by a light, but thick, core. As a material for the face layers, laminates of fiber-reinforced materials are widely used, but also sheet metal is applied. Different types of sandwich structures are distinguished depending on the design of the core, which could be homogeneously supported (e.g. with a foam-like material), locally or regionally supported, or supported along specific spatial directions, possibly forming comb-like or prismatic cells. In Figure 1.1(a) [1] a segment of a honeycomb sandwich, i.e. a sandwich core with cell

[1] Throughout this work (a) and (b) will denote the left and right images in a figure wherever applicable.

walls forming prismatic hexagonal cells, is illustrated in an exploded view, while in Figure 1.1(b)s some other sandwich core layouts are illustrated in a top-down view. The cell walls often consist of very thin aluminum foil, while the cells themselves are sometimes additionally filled with foams for added strength. In this work, sandwich structures with directionally supported or comb-like core structures are specifically investigated. For both the face layers and the cell walls, isotropic materials such as aluminum sheets or foils are assumed.

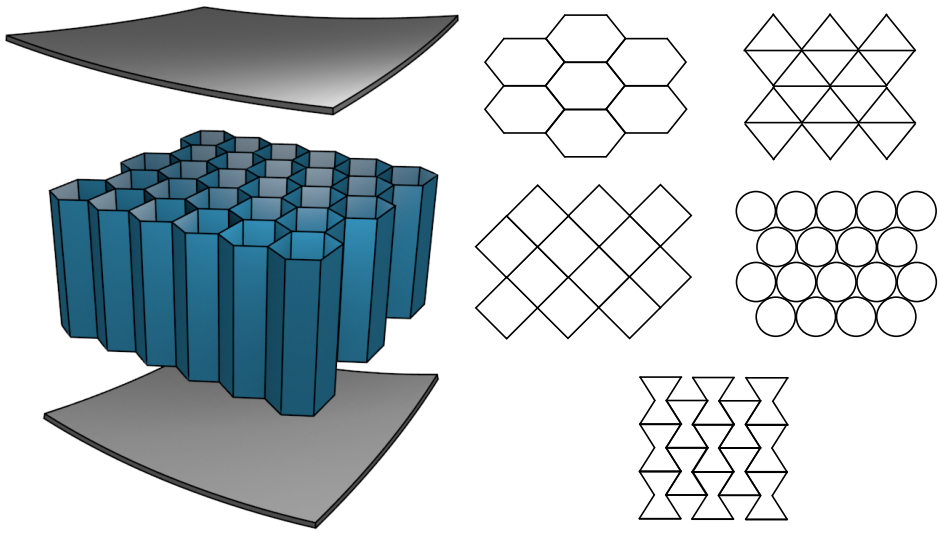


Figure 1.1: (a) Constituents (upper face layer, core, lower face layer) of a sandwich composite with hexagonal core structure, drawn separated from each other.
(b) Some types of sandwich core layouts in a top-down view.

The increasing use of composite materials in construction results in a demand for simulation algorithms which are able to predict the mechanical behavior of structural elements and whole structures consisting of composites. Even with ongoing advances in computational power, modeling complex composite structures with classical methods, such as a standard finite element model describing the full microstructure, leads in most cases to unreasonably large numerical systems. These can not be solved in a satisfactory amount of time, or sometimes not be solved at all due to enormous memory requirements. This is especially true for nonlinear models, where the resulting equation system must be solved many times during the course of a single simulation. It is thus expedient to develop new models and algorithms, purpose-built to accurately but also efficiently describe the behavior of a given complex system.

One classic means to do so is the homogenization approach, aiming to find effective elastic properties of a composite material by analytical or numerical means, and then modeling a simpler system with a homogeneous, but orthotropic, material utilizing these effective properties. This usually leads to drastically decreased calculation times and, in the linear case, to results with a high level of agreement to a full scale solution. However, the classical homogenization method assumes that the homogenized effective properties remain constant. Thus, the approach fails as soon as nonlinear system behavior is introduced, such as a non-elastic constitutive law or local geometric nonlinearities, e.g. consideration of stability modes such as local buckling. While failure mode estimates for various cases have been established in the literature, the post-critical behavior of a composite structure can in most cases not be described.

This work aims to present a method which is called the coupled global-local model, in which a homogenized global scale finite element model is combined with one local scale model in each of its integration points, so that every element is homogenized individually, depending on its current deformation state. The global-local model can not only initially determine effective elastic properties for a given sandwich cell microstructure, but also automatically re-evaluate the local properties that might be subject to change during the course of a nonlinear simulation. Nonlinear behavior can be described by determining changes to the microstructure and resulting effective properties on the fly, depending on the local deformation state in each integration point of the global finite element model. The ability to describe geometrically nonlinear behavior is especially valuable for shell structures, which can suffer from dramatic reductions in stiffness under certain stability modes. In addition, it is possible to investigate local behavior, such as the stress state in the microstructure in an approximate form or post-critical buckling modes, by pausing the simulation and zooming in to a specific local model, residing in a certain integration point of the global model. At the same time, the equation system can be significantly reduced in size, when compared to a classical full scale finite element model, and also be decoupled in large parts, allowing for faster solution times.

1.2 Overview

The global-local method presented in this work pursues an approach related to the classical homogenization: By evaluating a finite element model of a so-called representative volume element (RVE), describing the microstructure of a comb-like or directionally stiffened sandwich shell, effective elastic properties are determined by means of linearized stress resultants and stress resultants for a shell formulation. The RVE comprises a segment of the sandwich panel going through the full thickness, including not only the

sandwich core but also upper and lower face layers. However, with respect to the lateral directions, the RVE represents only a small part of the full structure, which is a prerequisite for the micro-mechanical homogenization approach. This RVE model will be referred to as the local scale model, which is implemented as a material law for a global scale model, in which the structure is modeled in its full spatial domain using standard shell elements. Hence, a Reissner-Mindlin shell formulation is used on both scales. A Hu-Washizu approach will be employed in order to prevent the occurrence of locking effects and to stabilize nonlinear simulations. In the case of linear behavior, the proposed method does in fact correspond to the classical homogenization method. The material response of the RVE is then evaluated only once for the whole system, instead of once in each integration point of every shell element on the global scale. However, in contrast to the classical homogenization method, the coupled model is extended to describe nonlinear behavior by re-evaluating the local systems, now one per integration point per global element, in each iteration and load step of a nonlinear simulation.

Compared to a classical full scale shell model depicting the whole microstructure of a composite sandwich material, the coupled method leads to substantially smaller equation systems in order to achieve a converged result. In addition to reducing the number of total equations, the equation system actually degenerates into a large number of small equation systems, one for each RVE, plus a comparably small system for the global scale. While the solution of the global scale system depends on the results of the local systems, all local RVE equation systems are decoupled from each other. This leads to much lower computational cost, as solving a sparse linear system with N equations still requires an effort of $\mathcal{O}(N^2)$, which is evidently larger than $\mathcal{O}(N_1^2) + \ldots + \mathcal{O}(N_n^2)$ for $N = N_1 + \ldots + N_n$. Moreover, the decoupled structure allows to solve the local RVE equation systems in parallel, perfectly suited to the hardware architecture of modern computers.

The coupled model is implemented in the FEM program FEAP [51], a Fortran code relatively popular in the scientific and research environment. In order to evaluate the local boundary value problems, a new instance of FEAP is started as a separate process in every integration point, once per iteration in a nonlinear simulation. This allows most existing program structures to be reused and also opens the possibility to load and investigate the local systems as input files separately, when a coupled simulation is finished or paused. In this way, it is possible to visualize the local stress state or the deformation of the microstructure, including the occurrence of buckling phenomena. On the downside, the frequent starting of new processes introduces a non-negligible amount of computational overhead, leading to non-optimal calculation times. Using certain measures such as a direct data transfer via the shared system memory or the explicit output and reutilization

of the local system assembly helps to somewhat reduce the occurring overhead.

Depending on the size of the models and discretization parameters, the coupled method in its existing implementation is in some cases able to significantly beat a full scale simulation in terms of computational time for nonlinear systems, even despite the introduced overhead, while in other cases it is slower. In all cases, the memory requirements are much lower and evidently in linear models, where only one RVE must be evaluated, the calculation is extremely fast. With a more competitive performance-orientated implementation, the proposed coupled method will drastically outperform classical simulations, both in linear and nonlinear cases, due to the massive decoupling and the reduction in the total number of equations. However, this thesis represents a proof of concept for the coupled global-local model and does not aim at realizing the most effective implementation, for which FEAP would not represent the best framework anyways.

1.3 State of research

For the linear case, the deformation behavior and pre-critical stability behavior of sandwich structures has been covered extensively in the literature via experiments and with analytical and numerical homogenization methods. In contrast to that, for the nonlinear case and post-critical behavior, only very few results are present where other types of composite structures, such as layered composites, are investigated.

In the following overview only a small amount of works can be considered. For a more comprehensive list, various review articles are present e.g. for recent developments in shell formulations and elements [53, 98], or specifically for the modeling of sandwich structures [37, 58].

Experiments and analytical approaches to determine orthotropic elasticity properties for honeycomb sandwich panels have a longstanding history, mainly due to their common application in aerospace engineering. Bending behavior [41] and shear moduli [52] for sandwich cores have been investigated already in the 1950s. The latter work also mentions the effect of core face sheet constraint, which is not taken into account in many studies where only a sandwich core without face layers is investigated. In [99] the theoretical prediction of wrinkling stability modes is carried out. This topic has been further investigated in [96] relatively recently. An early study including an investigation of cell wall imperfections is present in [33]. Many textbooks concerned with the design, construction and theoretical properties of sandwich panels are available, such as the classic works from Allen [2] or Plantema [76], or more recent works [7, 50, 86, 100].

Analytical homogenization approaches for heterogeneous elastic materials using a multi-scale field expansion can be found e.g. in [61]. The special case of a honeycomb sandwich is one of the discussed examples. Similar approaches are made in [49, 50] for sandwich cores with quadrilateral cells. Computational approaches for periodic microstructures have been illustrated in [63].

Effective elastic properties for certain triangular [36], quadrilateral and hexagonal [38] core layouts, comparable to the ones investigated in this thesis, have been determined using an energy-based approach both analytically as well as numerically. A homogenization approach for sandwich composites with combs or filled with a foam has been proposed in [34, 35], with additional investigation of delamination effects and insight on how to optimize a sandwich structure. From the same author, a review paper [37] is present where many works related to determining effective stress-strain relations in cellular sandwich cores are discussed. In particular, closed expressions or upper and lower bounds for the effective elastic constants of honeycomb cores are presented. Further review papers concerned with modern approaches to modeling elastic sandwich panels have been written [11, 39].

Another linear homogenization technique is shown in [90], where orthotropic shell elements with effective stiffness parameters are used to model corrugated core sandwich (CCS) panels, as applied in vehicle frame structures such as metro vehicles or buses. Effective stiffness parameters are compared for stiffener placement in the sandwich core under varying angles to the face layers.

Various works are present which cover stability modes of composite sandwiches, mainly by deriving critical buckling stresses or loads. First experimental works again go back to the 1950s [71], with plenty more recent works being available [24, 101]. The stability of honeycomb sandwiches is investigated in [74], where analytical formulae for critical loads based on homogenization results are used to predict stability modes such as global buckling, facesheet wrinkling and cell wall buckling for a sandwich unit cell. This stability model is then embedded in a periodic finite element formulation.

An overview over computational models for sandwich structural elements is given in [70]. The general concept of computational homogenization and, based on that, FE^2 multiscale modeling of heterogeneous materials were developed and covered extensively in the last decades. By now, plenty of literature is available, of which some selected works will be briefly described in the following. An early work discussing the use of new variational principles for composite materials is given in [78]. In [20, 44] the variational multiscale method has been introduced as a general paradigm. Many different types of heterogeneous materials and composites, such as porous media, fiber-reinforced materials,

polycrystallines, and more, were investigated with the help of multiscale methods [18, 22, 23, 55, 56, 66, 91]. A multiscale approach to optimize the design of honeycomb sandwich plates is introduced in [12, 13], where first the core properties are homogenized by evaluating a finite element model of a representative volume element, which are then optimized using a genetic algorithm [68]. Foam-like sandwich cores are investigated with the help of nested multiscale simulations in [85].

Coupled non-linear models related to the method described in this work have already been proposed for other types of microstructures. In [27] a layered structure is considered, where a shell model on the global scale is combined with 3d brick elements for the RVE. The deformation obtained in the global system is imposed onto the RVE by means of in-plane displacements on the lateral boundaries just as in this work. A numerical multiscale model for layered sandwich panels based on a plate formulation and a multi-level Newton algorithm is proposed in [28], and is extended to account for thickness changes in [29], but suffers from having high computational cost. In [15, 16, 21] related approaches for structured thin sheets are outlined based on Kirchhoff-Love thin shell formulations. Here, the RVEs are discretized with brick elements. A fiber-reinforced laminate as a microstructure is investigated in [83], where additionally interlaminar shear stresses are evaluated. This work also describes how the element loop in the finite element algorithm can be parallelized in order to solve the local boundary value problems simultaneously for maximum efficiency.

This thesis is based mainly on the aforementioned work [27] and extends the concept of the coupled nonlinear global-local model to a new type of structures, namely sandwich panels. Some earlier results related to this work have been published [31].

1.4 Outline

Following the current introductory chapter, the theoretical fundaments which are required for the main part of this work – the development of of the coupled global-local shell model – are presented in a brief overview.

In Chapter 2, important basic equations of mechanics are derived for a generic continuum. Specifically, descriptions and measures for deformations, strains and stresses are introduced, allowing the formulation of conservation laws which in turn lead to equilibrium conditions. In addition to that, some principles regarding constitutive laws are presented, including an elasto-plastic material model well-suited for metals like aluminum, which acts as a reference material for later numerical examples. The chapter is concluded with

an introduction to variational and linearization principles required as a basis for a finite element formulation, as well as a motivation for mixed field formulations such as the Hu-Washizu approach in order to avoid locking phenomena.

In Chapter 3, the generic continuum theory is applied to the special case of shell structures, which will be considered in this work. This leads to a reformulation of field equations and balance laws by means of shell strains and stress resultants, specifically suited to the geometric properties of a shell. With the help of variational calculus, the weak form of equilibrium is derived for a displacement formulation as well as for the Hu-Washizu approach. Based upon these results, Chapter 4 briefly introduces the ensuing steps of discretization and linearization in order to be able to obtain a finite element formulation of the nonlinear boundary value problem. This can then be solved iteratively in a numerical simulation.

Chapter 5 uses all these preliminary considerations to introduce the main part of this work, the so-called coupled global-local shell model. Here, a separate local shell model for an RVE is set up to describe the behavior of the microstructure of a sandwich, serving as a constitutive law in each integration point of a global scale shell model. Since only shell elements are employed on both scales, no additional introduction of element formulations is required. Instead, this chapter details how the boundary conditions are set for the RVE, and how the local models contribute to the weak form of equilibrium and its linearization. Moreover, it is highlighted how the coupling between global and local systems is given and that large parts of the equation system are in fact decoupled of each other, allowing for a faster calculation. Related to that, the simultaneous iteration is introduced as an additional measure to be able to reduce calculation times in many cases.

In Chapter 6, several numerical example models for sandwich structures will be presented, utilizing the coupled global-local method introduced in the previous chapters. Simulation results will be compared to classical full scale shell models and in some of the simpler cases to analytical results. Three different types of microstructures will be investigated: 1) an unidirectionally stiffened sandwich, which can be seen as a parallel arrangement of multiple I-profiles and will be used to model a beam-like structure; 2) a bidirectionally stiffened structure with cell walls along the x- and y-directions, forming a grid-like shape in a top-down view; 3) a structure with a hexagonal honeycomb core. All examples start with an investigation of the matrix of RVE stress resultants, followed by taking a look at the behavior of a linear coupled system. For the uni- and bidirectionally stiffened structures, geometrical and physical nonlinearities will be added as well in order to show the full capabilities of the coupled method with respect to nonlinear systems.

After presenting the theoretical foundations and subsequently the coupled global-local

model and numerical examples in Chapters 2 – 6, the following chapters are devoted to some noteworthy aspects that came up when working with the proposed model in real life. Chapter 7 discusses some of the difficulties arising for the generation of meshes for the numerical examples, and how to overcome them. For this means, various scripts have been implemented using Python [79] and Matlab [92], either for automatically generating a complete finite element model as an input file for FEAP (for the uni- and bidirectionally stiffened structures), or for generating a unit cell (for the honeycomb structure) which can be assembled in the commercial toolkit Abaqus subsequently and then converted to a FEAP input file.

Some basic concepts and issues of parallel programming, applied to a finite element program, are presented in Chapter 8. In particular, a shared memory interface is presented, which was developed within the scope of this work in order to provide for a high-performance exchange of data between global and local models in a coupled simulation. For some of the larger examples in Chapter 6, several gigabytes of data had to be transferred between processes in each iteration step, making a quick way of data transfer very advantageous. The performance of the shared memory interface is evaluated in a benchmark and compared to the previously available implementation using hard disk transfer.

In the final chapter, Chapter 9, the work is concluded with a summary and some suggestions for future improvements.

Chapter 2

Basic principles of continuum mechanics

In this chapter, some important basic concepts of continuum mechanics will be introduced, which will later be used in application to shell formulations. These principles are required as a theoretical fundament for the finite element method. For a more comprehensive overview the reader is referred to numerously available standard textbooks [3, 5, 14, 40, 60].

In the theory of continuum mechanics, a macroscopic body is considered as a system of continually distributed material points. This system is referred to as a continuum body $\mathcal{B}$ with its boundary $\partial\mathcal{B}$. The continuum body is described in an Euclidean space for given times $t \geq t_0$ by identifying each material point with a unique position vector.

It is assumed in the following that indexes described by Roman letters $i, j, \ldots$ range from $1-3$, Greek indexes α, β from $1-2$ and that the summation convention is employed.

2.1 Reference configuration and current configuration

To describe a continuum body undergoing a deformation process, its state at the time $t = t_0$ is considered, which is called the reference configuration, identified by the continuum body $\mathcal{B}_0$ and its boundary $\partial\mathcal{B}_0$. A position vector $\mathbf{X}$, with respect to the origin of an arbitrary basis $\mathbf{e}_i$ $(i = 1, 2, 3)$, can be assigned to each material point in the reference configuration. In a later point in time $t > t_0$, the so-called current configuration describes the continuum body $\mathcal{B}_t$ and its boundary $\partial\mathcal{B}_t$, both of which are now possibly deformed. The material points in the deformed configuration are identified by position vectors $\mathbf{x}$ with respect to the origin of a basis. In this work, the bases in reference configuration and current configuration will be identical and $t_0 := 0$ is chosen without any loss of generality.

To be able to correlate the state of a continuum body in the reference configuration and

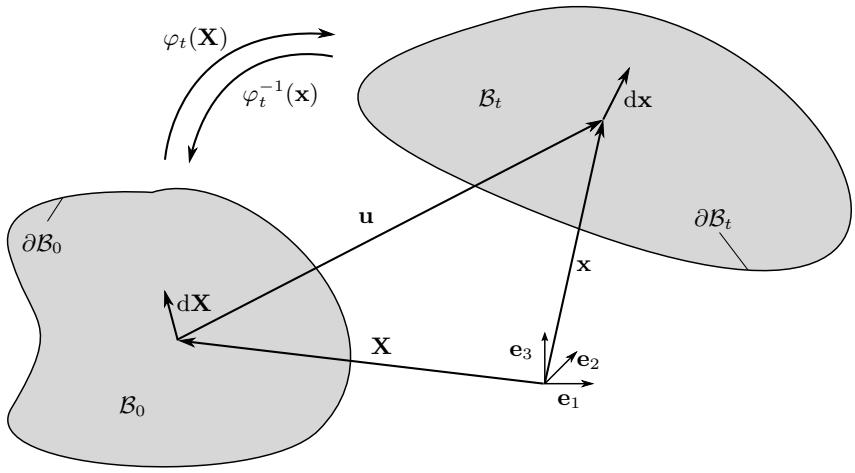


Figure 2.1: Reference configuration and current configuration of a continuum body.

the current configuration, the mapping

$$\hat{\varphi} : \mathcal{B}_0 \times \mathbb{R}^+ \to \mathcal{B}_t, \quad (\mathbf{X}, t) \mapsto \mathbf{x} \tag{2.1}$$

is introduced, which maps the position vector $\mathbf{X}$ of a material point in the reference configuration and a given time t onto the corresponding position vector $\mathbf{x}$ of the same material point in the current configuration at the time t. The mapping $\hat{\varphi}$ is bijective, each material point of the continuum $\mathcal{B}_t$ in the current configuration is the image of exactly one point in the reference configuration $\mathcal{B}_0$. Thus, the inverse of $\hat{\varphi}$ can be found. It is introduced as $\hat{\varphi}^{-1}$ and maps the each material point in the current configuration, identified by a position vector $\mathbf{x}$ and the current time t, onto the position vector $\mathbf{X}$ of that same material point in the reference configuration.

$$\hat{\varphi}^{-1} : \mathcal{B}_t \times \mathbb{R}^+ \to \mathcal{B}_0, \quad (\mathbf{x}, t) \mapsto \mathbf{X} \tag{2.2}$$

The mappings $\hat{\varphi}(\mathbf{X}, t)$ and $\hat{\varphi}^{-1}(\mathbf{x}, t)$ are often written in a more compact way as $\varphi_t(\mathbf{X})$ and $\varphi_t^{-1}(\mathbf{x})$.

In the special case of shell structures, which will be predominantly considered in this work, it is convenient to parametrize the continuum body in special curvilinear, so-called convective coordinates ξ^1, ξ^2, ξ^3. By that means, the convective coordinate lines are deformed in conjunction with the continuum body. The tangent vectors to the coordinate lines form a covariant basis in this case, which can be described in the reference

configuration and current configuration with:

$$\mathbf{G}_i = \frac{\partial \mathbf{X}}{\partial \xi^i} \quad \text{and} \quad \mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi^i} \tag{2.3}$$

The corresponding reciprocal vectors will then form a contravariant basis, they can be obtained by means of the orthogonality conditions

$$\mathbf{G}^i \cdot \mathbf{G}_j = \delta_{ij} \quad \text{and} \quad \mathbf{g}^i \cdot \mathbf{g}_j = \delta_{ij} \quad , \tag{2.4}$$

where δ_{ij} represents the Kronecker delta. Using this, the covariant coefficients G_{ij} and g_{ij} can be derived from the scalar products of the associated basis vectors.

After introducing the fundamental description of a continuum body in the reference configuration and current configuration, additional measures can now be provided to specify the possible deformation of a body.

2.2 Deformation and strains

Let each material point of a continuum body be identified by its position vector $\mathbf{X}$ in the reference configuration. When the body is subjected to a deformation, each of these points incurs a displacement $\mathbf{u}$ and finds itself in a new position $\mathbf{x}$.

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \tag{2.5}$$

The deformation process is uniquely determined by the displacement vector $\mathbf{u}$, defined in each point by

$$\mathbf{u} = \mathbf{x} - \mathbf{X} \quad . \tag{2.6}$$

The position coordinates x_i of the deformed body are called Eulerian coordinates, while the undeformed coordinates X_i are called Lagrangian coordinates. Specifying the deformation of a continuum body in the current configuration or the reference configuration is called Eulerian and Lagrangian description, respectively.

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) \text{ or } \mathbf{X} = \mathbf{X}(\mathbf{x}, t) \tag{2.7}$$

2.2.1 Deformation gradient

Applying the mapping $x_i = x_i(X_j)$ onto a line element in the reference configuration $\mathrm{d}\mathbf{X}$ with components $\mathrm{d}X_j$ leads to a deformed line element $\mathrm{d}\mathbf{x}$. Its components $\mathrm{d}x_i$ are then

given by evaluating the total derivative

$$\mathrm{d}x_i = \frac{\partial x_i}{\partial X_j}\,\mathrm{d}X_j \quad . \tag{2.8}$$

The tensor-valued deformation gradient $\mathbf{F}$ is introduced, containing the derivatives of all deformed position coordinates x_i with respect to the coordinates X_j in the reference configuration.

$$\mathbf{F} = \mathrm{Grad}\,\mathbf{x} := \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \quad \text{or} \quad [F_{ij}] = \left[\frac{\partial x_i}{\partial X_j}\right] \tag{2.9}$$

The deformation gradient maps the line element $\mathrm{d}\mathbf{X}$ in the reference configuration onto the deformed line element $\mathrm{d}\mathbf{x}$, it serves as an important measure to specify a body's deformations.

$$\mathrm{d}\mathbf{x} = \mathbf{F}\,\mathrm{d}\mathbf{X} \quad \text{or} \quad \mathrm{d}x_i = F_{ij}\mathrm{d}X_j \tag{2.10}$$

When using a convective coordinate system, a representation according to the covariant and contravariant basis vectors can be suitable:

$$\mathbf{F} = \mathbf{g}_i \otimes \mathbf{G}^i \tag{2.11}$$

As the descriptions in the reference configuration and the current configuration can be uniquely inverted, the deformation gradient's inverse mapping $\mathbf{F}^{-1}$ exists, it maps a deformed line element in the Eulerian specification back onto the original material line element in the reference configuration.

$$\mathrm{d}\mathbf{X} = \mathbf{F}^{-1}\,\mathrm{d}\mathbf{x} \quad \text{with} \quad \mathbf{F}^{-1} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \tag{2.12}$$

Due to the invertibility and the additional requirement that the continuum body must not penetrate itself during a deformation process, the deformation gradient's Jacobian must be positive.

$$J = \det \mathbf{F} = \det\left[\frac{\partial x_i}{\partial X_j}\right] > 0 \tag{2.13}$$

In addition, J is a measure for the relative change of volume in the course of a deformation. An infinitesimal volume $\mathrm{d}V$ will be mapped by the Jacobian J onto the deformed volume $\mathrm{d}v$:

$$\mathrm{d}v = J\mathrm{d}V \tag{2.14}$$

To describe the deformation process of an area element $\mathrm{d}\mathbf{A} = \mathrm{d}A\mathbf{n}_0$, represented by the area A and the unit normal vector $\mathbf{n}_0$ in the reference configuration, its mapping onto the corresponding deformed area element $\mathrm{d}\mathbf{a} = \mathrm{d}a\mathbf{n}$ can be obtained by

$$\mathrm{d}\mathbf{a} = J\mathbf{F}^{-T}\mathrm{d}\mathbf{A} \quad , \tag{2.15}$$

where $\mathbf{F}^{-T} = \left(\mathbf{F}^{-1}\right)^T$ represents the transposed inverse of $\mathbf{F}$.

2.2.2 Displacement gradient

According to the deformation gradient, the displacement gradient $\mathbf{H}$ is introduced by

$$\mathbf{H} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \quad \text{or} \quad [H_{ij}] = \left[\frac{\partial u_i}{\partial X_j}\right] \quad . \tag{2.16}$$

The displacement gradient can also be represented by means of the deformation gradient.

$$\mathbf{H} = \mathbf{F} - \mathbf{I} \tag{2.17}$$

Using the displacement gradient, the deformation of the previously considered line element can be described as follows:

$$\mathrm{d}\mathbf{u} = \mathrm{d}\mathbf{x} - \mathrm{d}\mathbf{X} = (\mathbf{F} - \mathbf{I})\,\mathrm{d}\mathbf{X} = \mathbf{H}\,\mathrm{d}\mathbf{X} \tag{2.18}$$

Here, $\mathbf{I}$ denotes the second order unit tensor.

2.2.3 Polar decomposition

It can be proven that the mapping $\mathbf{F}$ can be separated into a concatenation of a pure stretch tensor and a rotational tensor.
It holds

$$\mathbf{F} = \mathbf{R}\,\mathbf{U} = \mathbf{V}\,\mathbf{R} \tag{2.19}$$

where $\mathbf{R}$ is a proper orthogonal tensor ($\mathbf{R}^T = \mathbf{R}^{-1}, \det \mathbf{R} = 1$), called the rotation tensor, while $\mathbf{U}$ (right stretch tensor) and $\mathbf{V}$ (left stretch tensor) are symmetrical tensors. This multiplicative separation is uniquely determined and is commonly referred to as the right or left polar decomposition. The deformation of an arbitrary line element, $\mathrm{d}\mathbf{X} \mapsto \mathrm{d}\mathbf{x}$, can therefore be considered as a pure rotation, followed by a stretching, or vice versa.

2.2.4 Strain tensors

The deformation gradient as a measure for deformation is not invariant to rigid body motions. For instance, the rotation tensor $\mathbf{R}$ always represents a rigid body motion locally. $\mathbf{F}$ and $\mathbf{R}$ are thus unsuitable measures to describe strains. However, with help of the stretch tensors $\mathbf{U}$ and $\mathbf{V}$, several strain tensors can be introduced:

$$\begin{aligned}
\mathbf{U} &= \sqrt{\mathbf{F}^T\,\mathbf{F}} && \text{right stretch tensor}\\
\mathbf{V} &= \sqrt{\mathbf{F}\,\mathbf{F}^T} && \text{left stretch tensor}\\
\mathbf{C} &= \mathbf{U}^2 = \mathbf{F}^T\,\mathbf{F} && \text{right Cauchy-Green tensor}\\
\mathbf{B} &= \mathbf{V}^2 = \mathbf{F}\,\mathbf{F}^T && \text{left Cauchy-Green tensor}\\
\mathbf{E} &= \tfrac{1}{2}\left(\mathbf{F}^T\,\mathbf{F} - \mathbf{I}\right) && \text{Green-Lagrange strain tensor}\\
\boldsymbol{\gamma} &= \tfrac{1}{2}\left(\mathbf{I} - \mathbf{F}^{-T}\,\mathbf{F}^{-1}\right) && \text{Euler-Almansi strain tensor}
\end{aligned} \tag{2.20}$$

As a result of their definitions all these tensors are symmetrical. Additionally, some of them such as the Green-Lagrange strain tensor $\mathbf{E}$ are invariant to rigid body motions. $\mathbf{E}$ can be written using the displacement gradient:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{H}^T + \mathbf{H} + \mathbf{H}^T\,\mathbf{H}\right) \tag{2.21}$$

In this representation, the linear and nonlinear parts of the deformation are obviously separated from each other, which can be advantageous in the case of small strains. In terms of the so-called linear theory, higher order terms in $\mathbf{H}$ vanish and the Green-Lagrange strain tensor $\mathbf{E}$ and the Euler-Almansi strain tensor $\boldsymbol{\gamma}$ coincide to form the linearized strain tensor $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\varepsilon} = \mathbf{E} = \boldsymbol{\gamma} = \frac{1}{2}\left(\mathbf{H}^T + \mathbf{H}\right) \tag{2.22}$$

When using convective coordinates, the Green-Lagrange strain tensor is described by

$$\mathbf{E} = \frac{1}{2}\left(g_{ij} - G_{ij}\right)\mathbf{G}^i \otimes \mathbf{G}^j \quad . \tag{2.23}$$

2.3 Stress

2.3.1 Traction vector

When a continuum body is subjected to a deformation process, inner reaction forces and stresses are generated. They can be obtained by doing an imaginary cut along a plane surface in the interior of the body, which is specified by a position vector $\mathbf{x}$ and the outer

unit normal vector $\mathbf{n}$. The Cauchy traction vector $\mathbf{t}$ is then given by

$$\mathbf{t} = \mathbf{t}(\mathbf{x}, \mathbf{n}) = \lim_{A \to 0} \frac{\mathbf{f}}{A} \quad , \tag{2.24}$$

where $\mathbf{f}$ is the resulting force on the surface A in the chosen plane.

2.3.2 Stress tensors

As can be seen in (2.24), $\mathbf{t}$ in any point $\mathbf{x}$ depends on the spatial orientation $\mathbf{n}$ of the chosen plane. The state of stress can be described for any cut with arbitrary orientation when choosing three planes with pairwise orthogonal orientations. When specifically employing the base vectors $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ of the Euclidean space as outer normal vectors, the Cauchy stress tensor $\mathbf{T}$ is obtained:

$$\mathbf{T} = [T_{ij}] = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \tag{2.25}$$

As a result, $\mathbf{T}$ maps an arbitrary normal vector $\mathbf{n}$ onto the corresponding traction vector $\mathbf{t}$.

$$\mathbf{t} = \mathbf{T}\,\mathbf{n} \quad \text{or} \quad t_i = T_{ij} n_j \tag{2.26}$$

It can be shown, by evaluating the local balance of angular momentum, that the Cauchy strain tensor is always symmetrical in the static case.

$$\mathbf{T} = \mathbf{T}^T \quad \text{or} \quad T_{ij} = T_{ji} \tag{2.27}$$

Since the Cauchy strain tensor is introduced with respect to a position $\mathbf{x}$ in the current configuration, it describes the actually existing physical stress state in a continuum body. On the contrary, the nominal stress correlating to the reference configuration is given by the first Piola-Kirchhoff stress tensor $\mathbf{P}$, which can be derived from the Cauchy stress tensor:

$$\mathbf{P} = J\,\mathbf{T}\,\mathbf{F}^{-T} \tag{2.28}$$

Further stress tensors have been introduced, such as the Kirchoff stress tensor, which is given by $\boldsymbol{\tau} = J\,\mathbf{T}$ and acts in the current configuration, the second Piola-Kirchhoff stress tensor $\mathbf{S}$, which acts in the reference configuration, among several others. In the context of numerical simulations in solid mechanics, the second Piola-Kirchhoff stress tensor $\mathbf{S}$ is of particular importance, as it is symmetrical, unlike the first Piola-Kirchhoff stress

tensor $\mathbf{P}$, and it is work conform to the Green-Lagrange strain tensor $\mathbf{E}$ introduced in (2.20). $\mathbf{S}$ is given by

$$\mathbf{S} = J\,\mathbf{F}^{-1}\,\mathbf{T}\,\mathbf{F}^{-T} \tag{2.29}$$

and can be expressed using convective coordinates as follows:

$$\mathbf{S} = S^{ij}\,\mathbf{G}_i \otimes \mathbf{G}_j \tag{2.30}$$

Next to inner reaction forces and stresses, a continuum body is usually subjected to outer forces on parts of its boundary or its whole boundary. In addition, outer volume forces like gravitation can be present, taking effect on each material point inside of the continuum. These outer forces are generally causing the deformation process of a body.

2.4 Conservation laws

In the field of continuum mechanics, just as in most other disciplines of applied physics, a set of several balance laws is of fundamental importance to describe the state of a system. In this case, the conservation of mass, momentum and energy must be considered as well as the entropy inequality. The general proposition of a balance law is that the rate of change of a specific conserved quantity χ of a fixed system is determined by production or drain terms within the system and flow into or out of the system through its boundary.

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\chi = \int\limits_{\Omega} \frac{\partial \chi}{\partial t}\mathrm{d}v + \int\limits_{\partial\Omega} \chi\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}a \tag{2.31}$$

Here, the volume integral $\int_\Omega \frac{\partial \chi}{\partial t}\mathrm{d}v$ represents a production or drain term of the conserved quantity χ inside of the domain Ω. The integrand $\chi\mathbf{v}\cdot\mathbf{n}$ describes the ingoing and outgoing flow of χ, perpendicular to an infinitesimal boundary surface $\mathrm{d}a$ with outer unit normal vector $\mathbf{n}$. This term is integrated over the whole boundary of the domain $\partial\Omega$.

2.4.1 Conservation of mass

A continuum body is conferred a mass by the set of material points contained in it. The mass does not change when a body undergoes movement and deformation. The law of mass conservation can be formulated according to the distinct states of reference

configuration and current configuration on a global scale, which yields:

$$m = \int_{\mathcal{B}_0} \rho_0(\mathbf{X})\,\mathrm{d}V = \int_{\mathcal{B}_t} \rho(\mathbf{x},t)\,\mathrm{d}v = \text{const} \tag{2.32}$$

Here, $\rho_0(\mathbf{X})$ describes the mass density of an infinitesimal volume $\mathrm{d}V$ in the reference configuration, $\rho(\mathbf{X},t)$ describes the current local mass density of an infinitesimal volume $\mathrm{d}v$ in the current configuration. If the mass density is independent of the position vectors $\mathbf{X}$ and $\mathbf{x}$, the continuum is called homogeneous. Considering (2.14),

$$J = \frac{\rho_0}{\rho} \tag{2.33}$$

can be obtained. The Jacobian J is thus not only a measure for the change of volume caused by a deformation, it describes the change of mass density as well.

2.4.2 Conservation of linear and angular momentum

The law of conservation of linear momentum in its global form reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{B}_0} \rho_0 \mathbf{v}_0\,\mathrm{d}V = \int_{\mathcal{B}_0} \mathbf{b}_0\,\mathrm{d}V + \int_{\partial\mathcal{B}_0} \mathbf{t}_0\,\mathrm{d}A \tag{2.34}$$

The balance law can be written in a simplified form as $\dot{\mathbf{L}} = \mathbf{f}_0$, it states that $\dot{\mathbf{L}}$, the derivative of linear momentum with respect to time, is identical to the resultant of all outer forces $\mathbf{f}_0$. This equation is also known as the law of motion and is a generalization in the terms of continuum mechanics of Newton's second law $\vec{F} = m\vec{a}$ in point mechanics. In (2.34), $\mathbf{v}_0$ denotes the velocity of an infinitesimal control volume, $\mathbf{b}_0$ an outer volume force or body force, and $\mathbf{t}_0$ an outer force vector acting upon the body's boundary. In the static case, $\dot{\mathbf{L}}$ vanishes and (2.34) is reduced to the equilibrium equation

$$\int_{\mathcal{B}_0} \mathbf{b}_0\,\mathrm{d}V + \int_{\partial\mathcal{B}_0} \mathbf{t}_0\,\mathrm{d}A = 0 \quad . \tag{2.35}$$

Using Cauchy's law (2.24) and the divergence theorem, the global equilibrium equation can be written as

$$\int_{\mathcal{B}_0} (\mathrm{Div}\,\mathbf{P} + \mathbf{b}_0)\,\mathrm{d}V = 0 \tag{2.36}$$

or, in its local form,

$$\mathrm{Div}\,\mathbf{P} + \mathbf{b}_0 = 0 \quad . \tag{2.37}$$

In the same way, the conservation law for angular momentum can be obtained. In the Lagrangian specification it reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{B}_0}\mathbf{r}\times\rho_0\mathbf{v}_0\,\mathrm{d}V=\int_{\mathcal{B}_0}\mathbf{r}\times\mathbf{b}_0\,\mathrm{d}V+\int_{\partial\mathcal{B}_0}\mathbf{r}\times\mathbf{t}_0\,\mathrm{d}A \qquad \text{or} \quad \dot{\mathbf{J}}=\mathbf{m}_0 \tag{2.38}$$

and states that the derivative of the angular momentum with respect to time $\dot{\mathbf{J}}$ equals the resultant external moment $\mathbf{m}_0$. In this case, the vector $\mathbf{r}$ denotes the position vector $\mathbf{r}=\mathbf{x}-\mathbf{x}_0$ with respect to a reference point $\mathbf{x}_0$. By evaluating the conservation law of angular momentum (2.38) locally, it can be shown that the Cauchy stress tensor $\mathbf{T}$ is symmetrical, and thus the second Piola-Kirchhoff stress tensor $\mathbf{S}$ is symmetrical as well.

2.4.3 Conservation of energy

The energy balance law is introduced restricted to mechanical and thermal energy terms only. This equation is also known as the first law of thermodynamics and can be written as follows:

$$\dot{\mathcal{K}}+\dot{\mathcal{E}}=\mathcal{P}_{\mathrm{ext}}+\mathcal{Q} \tag{2.39}$$

The law states that the change of global energy in a system with respect to time, which consists of kinetic energy $\mathcal{K}$ and internal energy $\mathcal{E}$, equals the resultant external power, consisting of a mechanical power $\mathcal{P}_{\mathrm{ext}}$ resulting from boundary forces and body forces, as well as the thermal power $\mathcal{Q}$. By inserting these quantities in their explicit forms, one can obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{B}_0}\left(\frac{1}{2}\rho_0\mathbf{v}_0\cdot\mathbf{v}_0+e_0\right)\,\mathrm{d}V=\int_{\mathcal{B}_0}\left(\mathbf{b}_0\cdot\mathbf{v}_0+r_0\right)\,\mathrm{d}V+\int_{\partial\mathcal{B}_0}\left(\mathbf{t}_0\cdot\mathbf{v}_0-\mathbf{q}_0\cdot\mathbf{n}_0\right)\,\mathrm{d}A \tag{2.40}$$

Here, e_0 denotes the internal mechanical energy per unit volume in the reference configuration, r_0 constitutes a heat source, $\mathbf{q}_0$ the heat flow, and $\mathbf{n}_0$ describes the unit normal vector on the boundary in the reference configuration.

If the thermal power is neglected, $\mathcal{Q}=0$, the rate of change of the internal mechanical energy is equal to the stress power $\mathcal{P}_{\mathrm{int}}$. Here, the rate of change $\dot{W}$ of the internal mechanical work is represented, which can be expressed using pairs of work conform stress and strain tensors, such as the second Piola-Kirchhoff stress tensor $\mathbf{S}$ and the Green-Lagrange strain tensor $\mathbf{E}$.

$$\dot{W} = \mathcal{P}_{\text{int}} = \dot{\mathcal{E}} = \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{B}_0} e_0 \, \mathrm{d}V = \int_{\mathcal{B}_0} \mathbf{P} : \dot{\mathbf{F}} \, \mathrm{d}V = \int_{\mathcal{B}_0} \mathbf{S} : \dot{\mathbf{E}} \, \mathrm{d}V \tag{2.41}$$

2.4.4 Entropy inequality

The entropy inequality, also known as the second law of thermodynamics or the Clausius-Duhem inequality, is relevant in so-called dissipative systems, where mechanical energy is converted into thermal energy. Introducing s_0 as the local entropy per unit volume, the global entropy of a continuum in the reference configuration can be obtained by integrating over the whole volume:

$$\mathcal{S} = \int_{\mathcal{B}_0} s_0 \, \mathrm{d}V \tag{2.42}$$

The second law of thermodynamics can then be written as

$$\Gamma = \dot{\mathcal{S}} - \tilde{\mathcal{Q}} \geq 0 \quad , \tag{2.43}$$

stating that the global production rate of entropy Γ must not be negative. The production rate is comprised of the difference of the rate of entropy $\dot{\mathcal{S}}$ and the rate of ingoing entropy $\tilde{\mathcal{Q}}$, which in turn consists of entropy flow through the system boundaries and entropy production inside of the system. Consequently, the entropy inequality determines the direction of energy exchange, it specifies, for instance, that heat is always transferred from warmer to colder areas inside of a continuum. For $\Gamma = 0$ the exchange process is reversible, whereas for $\Gamma > 0$ the energy is dissipated and the process cannot be reversed.

2.5 Constitutive laws

Up to now, the states of strain and stress inside of a continuum body have been considered isolated from each other. To be able to correlate strains and stresses, it is necessary to introduce constitutive equations, describing the material response of a deformable continuum.

Constitutive laws describe the physical behavior of a material that is observed on a macroscopic scale. The derivation of appropriate laws is an important research area for many materials used in practice. In particular, dissipative and dissipation-free constitutive laws can be distinguished. In the latter case, the mechanical strain energy is always conserved, hence any deformation process is reversible. Materials described by these laws

are also known as hyperelastic materials, they have a uniquely invertible correlation between stresses and strains. In the simplest case, stresses and strains are proportional to each other and the material is said to be linear-elastic.

2.5.1 Helmholtz free energy

In the special case of hyperelastic materials, it is possible to introduce the so-called Helmholtz free energy Ψ as a potential. Ψ is alternatively often referred to as the strain-energy function. It depends on the deformation gradient $\mathbf{F}$ or another suitable measure of deformation, such as the right Cauchy-Green tensor $\mathbf{C}$ or the left Cauchy-Green tensor $\mathbf{B}$, and must fulfill certain requirements, the most important ones of which will be discussed below.

The growth condition has to be satisfied, stating that for an imaginary infinite expansion or compression of a continuum, the required strain energy has to be infinitely large:

$$\Psi \to \infty \quad \text{for } J \to 0 \text{ or } J \to \infty \tag{2.44}$$

Additionally, in the undeformed state, all stresses must vanish, while during a deformation the strain energy must become positive.

$$\Psi(\mathbf{F} = \mathbf{I}) = 0 \tag{2.45}$$

$$\Psi(\mathbf{F}) > 0 \quad (\mathbf{F} \neq \mathbf{I}) \tag{2.46}$$

It is assumed that in homogeneous materials, the Helmholtz free energy is only dependent on the state of deformation, $\Psi = \Psi(\mathbf{F})$. The material behaves the same in all spatial locations. However, in a heterogeneous material an additional dependency on the position is given, $\Psi = \Psi(\mathbf{X}, \mathbf{F})$, the material behavior differs in each point.
A further distinction can be made by observing the material behavior along different spatial directions, leading to the classification of isotropic and anisotropic materials. If any deformation process can be superimposed by a rigid body motion in the reference configuration, without changing the Helmholtz free energy, the material is called isotropic, it behaves the same for all spatial directions. Otherwise, the material is called anisotropic if the Helmholtz free energy changes.

A general requirement for any constitutive law and thus a requirement for the Helmholtz free energy as well is the invariance with respect to rigid body motions. Let $\mathbf{Q}$ be an orthogonal tensor, representing an arbitrary rigid body motion. Then, for homogeneous

or heterogeneous materials

$$\Psi(\mathbf{F}) = \Psi(\mathbf{QF}) \quad \text{or} \quad \Psi(\mathbf{X}, \mathbf{F}) = \Psi(\mathbf{X}, \mathbf{QF}) \tag{2.47}$$

must hold, respectively. By means of the polar decomposition (2.19), Ψ must be independent of the rotation tensor $\mathbf{R}$, so that (2.47) can also be written as:

$$\Psi(\mathbf{F}) = \Psi(\mathbf{U}) \quad \text{or} \quad \Psi(\mathbf{X}, \mathbf{F}) = \Psi(\mathbf{X}, \mathbf{U}) \tag{2.48}$$

Using the first Piola-Kirchhoff stress tensor $\mathbf{P}$, the reaction stresses as a material response of a system exposed to a deformation process $\mathbf{F}$ can be expressed. In this representation, the character of the Helmholtz free energy as an energy potential becomes evident.

$$\mathbf{P}(\mathbf{F}) = \frac{\partial \Psi(\mathbf{F})}{\partial \mathbf{F}} \quad \text{or} \quad \mathbf{P}(\mathbf{X}, \mathbf{F}) = \frac{\partial \Psi(\mathbf{X}, \mathbf{F})}{\partial \mathbf{F}} \tag{2.49}$$

For this reason, in homogeneous materials, the stress tensor $\mathbf{P}$ is a function of the deformation process, represented by the deformation gradient $\mathbf{F}$ or any strain tensor. In the case of a heterogeneous material, an additional dependency with respect to the position $\mathbf{X}$ is given.

Assuming a homogeneous material, the different stress tensors can be derived by expressing the Helmholtz free energy to be dependent on suitable strain measures. For instance, the first Piola-Kirchhoff stress tensor $\mathbf{P}$ can alternatively be written by means of the right Cauchy-Green tensor $\mathbf{C}$:

$$\mathbf{P} = 2\,\mathbf{F}\,\frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}} \tag{2.50}$$

For the second Piola-Kirchhoff stress tensor $\mathbf{S}$, being work conform to the Green-Lagrange strain tensor $\mathbf{E}$,

$$\mathbf{S} = 2\,\frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}} = \frac{\partial \Psi(\mathbf{E})}{\partial \mathbf{E}} \tag{2.51}$$

is given, while the Cauchy stress tensor $\mathbf{T}$ can be written in the following way:

$$\mathbf{T} = J^{-1}\,\mathbf{F}\,\left(\frac{\partial \Psi(\mathbf{F})}{\partial \mathbf{F}}\right)^{T} = 2\,J^{-1}\,\mathbf{F}\,\left(\frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}}\right)\mathbf{F}^{T} \tag{2.52}$$

In the following, the Helmholtz free energy is assumed to be expressed as a function of the right Cauchy-Green tensor $\mathbf{C}$. It can be shown, that the dependency can be reduced to the three invariants of $\mathbf{C}$, so that by introducing

$$I_1 = \operatorname{tr}\mathbf{C}\,, \quad I_2 = \frac{1}{2}\left((\operatorname{tr}\mathbf{C})^2 - \operatorname{tr}\mathbf{C}^2\right)\,, \quad I_3 = \det\mathbf{C} \quad , \tag{2.53}$$

the Helmholtz free energy can be specified as:

$$\Psi = \Psi\left(I_1, I_2, I_3\right) \tag{2.54}$$

Applying the chain rule to the derivative $\partial\Psi(\mathbf{C})/\partial\mathbf{C}$ from (2.51), the second Piola-Kirchhoff stress tensor can be written by means of the Helmholtz free energy as a function of the invariants of $\mathbf{C}$:

$$\mathbf{S} = 2\left[\left(\frac{\partial\Psi}{\partial I_1} + I_1\frac{\partial\Psi}{\partial I_2}\right)\mathbf{I} - \frac{\partial\Psi}{\partial I_2}\mathbf{C} + I_3\frac{\partial\Psi}{\partial I_3}\mathbf{C}^{-1}\right] \tag{2.55}$$

2.5.2 J_2-plasticity for small strains

For many materials that are commonly used in technical applications, an elasto-plastic description of material behavior for small strains has proven to be effective. Particularly, this includes materials such as steel or aluminum, which is why an elasto-plastic material law will be used in some of the examples in this work. Only the basic theory for elasto-plastic models without hardening is shown here, for a more complete overview the reader is referred to e.g. [59, 89, 97].

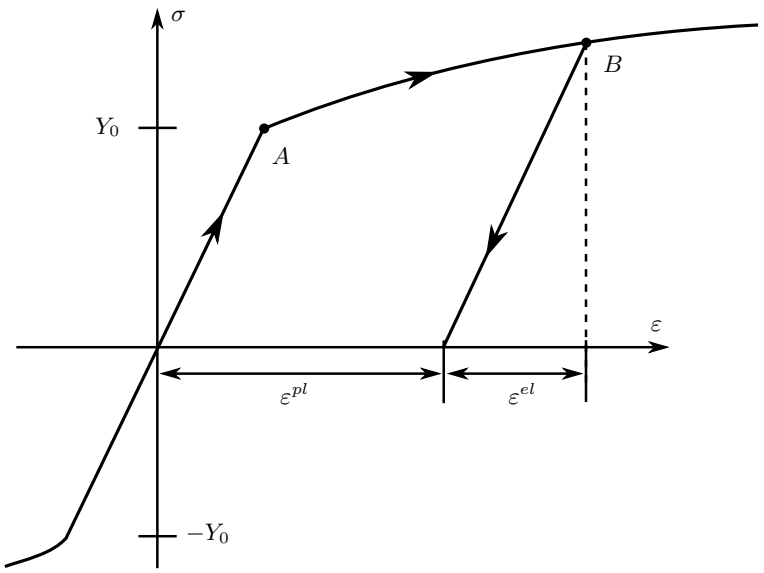


Figure 2.2: Typical stress-strain curve for a 1d tension test using an elasto-plastic material with hardening.

In the simplest case of a 1-dimensional tension test, system behavior as depicted in

Figure 2.2 is observed. The stress/strain-curve starts in the origin with a linear-elastic development up to the point A, where the yield stress Y_0 is reached. After that point, the slope of the curve drops as further strain increases lead to a smaller increase in stress. When unloading the structure in a point B, the stress and strain curve follows an affine path parallel to the linear-elastic segment between the origin and A. For $\sigma = 0$ an irreversible plastic strain ε^{pl} remains.

This concept is now transferred to the general 3d case for stresses and strains. With the premise of small strains, the strain tensor $\boldsymbol{\varepsilon}$ is split additively in an elastic and a plastic part:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + \boldsymbol{\varepsilon}^{pl} \tag{2.56}$$

For the elastic part of the deformation, the classical Hooke's constitutive law

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}^{el} \tag{2.57}$$

is employed.

Additionally, a yield condition must be introduced that specifies the extent of the elastic region of the deformation. It is formulated by means of an inequality constraint, depending on the stresses and, in the case of hardening, on the internal hardening variables. For the examples in this work, no hardening moduli where used, so that

$$F(\boldsymbol{\sigma}) = \phi(\boldsymbol{\sigma}) - Y_0 \leq 0 \tag{2.58}$$

can be introduced. Here, a structure undergoes an elastic deformation as long as $F(\boldsymbol{\sigma}) < 0$ and plastic deformations for $F(\boldsymbol{\sigma}) = 0$. Values $F(\boldsymbol{\sigma}) > 0$ can not be obtained.

The evolution of the flow process is described by a flow rule, which is given by

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \lambda \frac{\partial F}{\partial \boldsymbol{\sigma}} \quad . \tag{2.59}$$

Here, λ is a scalar describing the magnitude of the plastic strain increment in the direction of plastic flow, represented by the term $\dfrac{\partial F}{\partial \boldsymbol{\sigma}}$.

In order to define the yield condition for the case of J_2-plasticity, the stress deviator

$$\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\sigma})\,\mathbf{I} \tag{2.60}$$

is introduced. The yield condition is then postulated to depend only on the deviatoric

part of the stress state and is written as

$$\phi(\mathbf{s}) = \sqrt{\frac{3}{2}}\,|\mathbf{s}| = \sqrt{\frac{3}{2}\,\mathbf{s} : \mathbf{s}} \tag{2.61}$$

for isotropic materials.

The name J_2-plasticity thus stems from the fact that the yield condition only depends on the second invariant of the stress deviator $I_2(\mathbf{s}) = J_2$. This material model is also called von Mises plasticity.

In the space of principal stresses, the von Mises flow condition can be visualized as a cylinder with radius $\sqrt{\frac{2}{3}}\,Y_0$ around the hydrostatic axis $\{(\sigma_1, \sigma_2, \sigma_3)\,|\,\sigma_1 = \sigma_2 = \sigma_3\}$. Stress points in the interior of the cylinder represent an elastic state, while the cylinder surface corresponds to a state of plastic flow. Points outside of the cylinder are not admissible. In the case of a plane stress scenario ($\sigma_3 = 0$), the von Mises flow surface forms an ellipse in the σ_1, σ_2-plane.

It is important to mention that it is safe to make the assumption of small strains for the examples considered in this work, as well as usual technical applications of sandwich constructions in general.

2.6 Variational principles and linearization

In order to make mechanical equations accessible to computational methods, discretization techniques have to be employed. These techniques usually originate from weak forms of a field equation, which can be obtained by means of variational calculus.

2.6.1 Variational principle

Let the continuum body's boundary $\partial\mathcal{B}_0$ be separated into two disjoint partial boundaries $\partial\mathcal{B}_{0u}$ and $\partial\mathcal{B}_{0\sigma}$. On $\partial\mathcal{B}_{0u}$, the displacements $\mathbf{u}$ are prescribed, these boundary conditions are also known as displacement or kinematic boundary conditions or, more generally, Dirichlet boundary conditions, explicitly specifying the value of the field quantity. On the other hand, stress boundary conditions (also called static boundary conditions or Neumann boundary conditions) are given on $\partial\mathcal{B}_{0\sigma}$.

The local form of the static balance of linear momentum (2.37), relating to the reference configuration, is multiplied by an arbitrary feasible vector field, which is also known as

the test function. For this means, the virtual displacement field $\delta\mathbf{u}$ is introduced, which is independent of the actual displacements $\mathbf{u}$ and represents an imaginary infinitesimal displacement. The variational operator δ is linear and complies to the usual calculation rules of differentiation, just as the differential operator ∂. Feasibility requires compliance only with the displacement boundary conditions given by the boundary value problem, i.e. $\delta\mathbf{u} = 0$ on $\partial\mathcal{B}_{0u}$.

By integrating over the whole continuum body $\mathcal{B}_0$, the weak form of equilibrium is obtained:

$$g(\mathbf{u}, \delta\mathbf{u}) := \int_{\mathcal{B}_0} (-\operatorname{Div}\mathbf{P} - \mathbf{b}_0) \cdot \delta\mathbf{u} \, \mathrm{d}V = 0 \tag{2.62}$$

Using the fundamental lemma of calculus of variations, it can be seen that the weak form (2.62) is in fact equivalent to the balance of linear momentum (2.37) - the strong form of the boundary-value problem - since an arbitrary virtual displacement field $\delta\mathbf{u}$ can be chosen and thus the factor $(-\operatorname{Div}\mathbf{P} - \mathbf{b}_0)$ must vanish identically in $\mathcal{B}_0$.

With the help of integration by parts, the derivative of $\mathbf{P}$ can be transferred to the virtual displacements $\delta\mathbf{u}$, application of the divergence theorem then yields

$$g(\mathbf{u}, \delta\mathbf{u}) = \int_{\mathcal{B}_0} (\mathbf{P} : \operatorname{Grad}\delta\mathbf{u} - \mathbf{b}_0 \cdot \delta\mathbf{u}) \, \mathrm{d}V - \int_{\partial\mathcal{B}_{0\sigma}} \bar{\mathbf{t}}_0 \cdot \delta\mathbf{u} \, \mathrm{d}A = 0 \quad . \tag{2.63}$$

Here, $\bar{\mathbf{t}}_0$ denotes the prescribed traction vector on the boundary $\partial\mathcal{B}_{0\sigma}$, where the boundary condition reads $\mathbf{t}_0 = \mathbf{P}\mathbf{N} = \bar{\mathbf{t}}_0$ with outer unit normal vector $\mathbf{N}$. As the virtual displacements $\delta\mathbf{u}$ vanish on $\partial\mathcal{B}_{0u}$, the integration by parts yields no additional terms along that boundary.

Using the linearity of the operator δ,

$$\delta\mathbf{F} = \delta(\operatorname{Grad}\mathbf{u} + \mathbf{I}) = \delta(\operatorname{Grad}\mathbf{u}) = \operatorname{Grad}\delta\mathbf{u} \tag{2.64}$$

is obtained.
Consequently, the weak form (2.63) can be written as follows:

$$g(\mathbf{u}, \delta\mathbf{u}) := \int_{\mathcal{B}_0} (\mathbf{P} : \delta\mathbf{F} - \mathbf{b}_0 \cdot \delta\mathbf{u}) \, \mathrm{d}V - \int_{\partial\mathcal{B}_{0\sigma}} \bar{\mathbf{t}}_0 \cdot \delta\mathbf{u} \, \mathrm{d}A = 0 \tag{2.65}$$

This equation is known as the principle of virtual work. The summands in (2.65) can be separated into the inner virtual work δW_{int}, which can also be expressed using the

second Piola-Kirchhoff stress tensor,

$$\delta W_{\text{int}} = \int_{\mathcal{B}_0} \mathbf{P} : \delta\mathbf{F} \, \mathrm{d}V = \int_{\mathcal{B}_0} \mathbf{S} : \delta\mathbf{E} \, \mathrm{d}V \quad , \tag{2.66}$$

as well as the external virtual work δW_{ext},

$$\delta W_{\text{ext}} = \int_{\mathcal{B}_0} \mathbf{b}_0 \cdot \delta\mathbf{u} \, \mathrm{d}V + \int_{\partial\mathcal{B}_{0\sigma}} \bar{\mathbf{t}}_0 \cdot \delta\mathbf{u} \, \mathrm{d}A \quad . \tag{2.67}$$

In the static case, internal and external virtual work must be equal:

$$g(\mathbf{u}, \delta\mathbf{u}) = \delta W_{\text{int}} - \delta W_{\text{ext}} = 0 \tag{2.68}$$

In many applications of static continuum mechanics, it is safe to assume that the external loads do not change in direction or value during a deformation process. These loads are referred to as conservative loads, they always possess an energy potential Π when using hyperelastic materials.

While this circumstance is not necessary for the fulfillment of the principle of virtual work – which holds for all materials as well as for non-conservative loads – the existence of a potential can be exploited when designing numerical methods for the solution of the boundary value problem. For instance, they always lead to a symmetrical stiffness matrix in the context of the finite element method. Due to this reason, assuming the existence of a potential is often advantageous.

It is noted that in the case of inelastic materials, such as the elasto-plastic material introduced in Section 2.5.2, a potential formulation can not be used. Adjusted formulations for these cases can be found in the literature as well as in later chapters of this work for a shell formulation. For the rest of this chapter, the existence of an energy potential is presumed.

Inner and outer potential energy Π_{int} and Π_{ext} are introduced as follows:

$$\Pi_{\text{int}} = \int_{\mathcal{B}_0} \Psi \, \mathrm{d}V \tag{2.69}$$

$$\Pi_{\text{ext}} = -\int_{\mathcal{B}_0} \mathbf{b}_0 \cdot \mathbf{u} \, \mathrm{d}V - \int_{\partial\mathcal{B}_{0\sigma}} \bar{\mathbf{t}}_0 \cdot \mathbf{u} \, \mathrm{d}A \tag{2.70}$$

In this equation, Ψ represents the Helmholtz free energy function, depending on the deformation gradient $\mathbf{F}$ or any strain tensor, as shown in Section 2.5.1. A state of equilibrium can now be obtained by evaluating the directional derivative of the total

potential $\Pi = \Pi_{\text{int}} + \Pi_{\text{ext}}$ with respect to the displacements $\mathbf{u}$:

$$\delta\Pi(\mathbf{u}, \delta\mathbf{u}) = \mathrm{D}\left[\Pi(\mathbf{u})\right] \cdot \delta\mathbf{u} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Pi\left(\mathbf{u} + \varepsilon\delta\mathbf{u}\right)\big|_{\varepsilon=0} = 0 \tag{2.71}$$

Here, ε is a scalar and D denotes the differential operator according to the Gâteaux directional derivative. The variational principle (2.71) expresses that the derivative of the total potential must vanish with respect to any direction $\delta\mathbf{u}$. Applying the directional derivative onto the principle of virtual work (2.65), it can be shown that the potential energy Π is stationary for arbitrary variations $\delta\mathbf{u}$:

$$\delta\Pi(\mathbf{u}, \delta\mathbf{u}) = \delta\Pi_{\text{int}} + \delta\Pi_{\text{ext}} = \delta W_{\text{int}} - \delta W_{\text{ext}} = 0 \tag{2.72}$$

As the strain energy is in fact minimized in elastic materials, this equation is also known as the principle of minimum strain energy. In the general case, the second variation of the potential $\delta^2\Pi$ determines whether the stationarity point represents a maximum, minimum or saddle point. Introducing $\Delta\mathbf{u}$ as the increment of the displacement field, it follows using (2.71):

$$\delta^2\Pi = \delta^2\Pi(\mathbf{u}, \delta\mathbf{u}, \Delta\mathbf{u}) = \mathrm{D}[\delta\Pi(\mathbf{u}, \delta\mathbf{u})] \cdot \Delta\mathbf{u} \tag{2.73}$$

2.6.2 Linearization

In the general case, equation (2.65) is nonlinearly dependent on the displacements $\mathbf{u}$ and can only be solved analytically in some special cases. Consequently, numerical methods based on the finite element method are employed to find approximate solutions for the nonlinear problem with arbitrary boundary conditions, employing linearized formulations. Using iterative algorithms such as the Newton method, which is popular for its simplicity and the property of local quadratic convergence, a sequence of linear systems is solved. It is necessary to provide a consistent linearization of the problem's quantities.

Evaluating the Taylor series expansion of the functional g up to the first order,

$$L[g(\mathbf{u}, \delta\mathbf{u})] := g(\mathbf{u}, \delta\mathbf{u}) + \mathrm{D}\left[\, g(\mathbf{u}, \delta\mathbf{u}) \,\right] \cdot \Delta\mathbf{u} \tag{2.74}$$

is obtained.

$\Delta\mathbf{u}$, just as $\delta\mathbf{u}$ previously, represents a virtual infinitesimal displacement field on the continuum. With conservative external loads the linearization of the external virtual work vanishes, $\mathrm{D}[\, \delta W_{\text{ext}}(\mathbf{u}, \delta\mathbf{u}) \,] \cdot \Delta\mathbf{u} = 0$. Since D is a linear operator, it is sufficient to consider the linearization of the internal virtual work, which can be expressed by means of

the second Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor as follows:

$$\mathrm{D}\,[\,g(\mathbf{u},\delta\mathbf{u})\,]\cdot\Delta\mathbf{u} = \mathrm{D}[\,\delta W_{\mathrm{int}}(\mathbf{u},\delta\mathbf{u})\,]\cdot\Delta\mathbf{u} = \int_{\mathcal{B}_0} [\,\mathbf{S}:\Delta\delta\mathbf{E} + \delta\mathbf{E}:\mathbb{C}:\Delta\mathbf{E}\,]\,\mathrm{d}V \tag{2.75}$$

In the general case,

$$\mathbb{C} = 2\,\frac{\partial\mathbf{S}}{\partial\mathbf{C}} = \frac{\partial\mathbf{S}}{\partial\mathbf{E}} \tag{2.76}$$

holds, where the material tangent $\mathbb{C} = [C_{ijkl}]$ is a tensor of rank four. For the special case of hyperelastic materials, where $\mathbf{S} = \mathbf{S}(\mathbf{E})$ and $\mathbf{E} = \mathbf{E}(\mathbf{u})$ are given by (2.51), $\mathbb{C}$ is called the elasticity tensor and can be determined by evaluating the second order derivative of the Helmholtz free energy:

$$\mathbb{C} = 2\,\frac{\partial\mathbf{S}}{\partial\mathbf{C}} = 4\,\frac{\partial^2\Psi(\mathbf{C})}{(\partial\mathbf{C})^2} \tag{2.77}$$

In this case, the symmetry characteristics of the stress and strain tensors are propagated. Additionally, due to the derivation by means of the Helmholtz potential Ψ where the order of derivatives may be exchanged, minor symmetries $C_{ijkl} = C_{jikl} = C_{ijlk}$ are present, so that $\mathbb{C}$ has at most only 21 instead of 81 independent components. If additional material symmetries are present, such as for orthotropic, transversal isotropic or fully isotropic materials, this number of independent components is further reduced, down to a minimum of two independent constants for isotropic materials.

The linearization (2.74) acts as the basis for the implementation of a numerical algorithm. In this algorithm, the term $g(\mathbf{u},\delta\mathbf{u})$ expresses the residuum, so that in each step of a Newton iteration the incremental displacements $\Delta\mathbf{u}$ are calculated such that $L[g(\mathbf{u},\delta\mathbf{u})] = 0$ holds. The newly found displacement increment is then applied and the iteration continues, until the residuum vanishes and a solution for $\mathbf{u}$ is found.

2.6.3 Mixed field formulations

Pure displacement formulations such as that based on the principle of virtual work can in some cases lead to numerical problems when used as a basis of numerical calculations with the finite element method. Ill-conditioned stiffness matrices and locking effects can occur, especially when using finite elements with low order ansatz functions in bending-dominated problems, such as many plate and shell applications. This leads to discretized systems which behave too stiff, even when using reasonably fine meshes.

Several measures were developed to be able to deal with locking effects. One possibility

is the approach of reduced integration, which can however lead to a loss of regularity and zero energy modes under certain boundary conditions. Another remedy is the usage of so-called mixed or hybrid formulations, where a variational principle is employed depending not only on the displacements, but on multiple field quantities instead. For instance, a two-field formulation is known as the Hellinger-Reissner variational principle, where stresses and displacements are considered as independent field quantities. The downside of this formulation is that an inverse form of the constitutive equations must be evaluated, which is numerically expensive and might even be impossible to find in some cases.

More commonly used in numerical simulations is a three-field formulation comprising displacements, stresses and strains as independent field quantities. This formulation is referred to as the Hu-Washizu variational principle [42, 95]. It can be interpreted as a more general form of the principle of virtual work and reads, see e.g. [40]:

$$\begin{aligned}\Pi(\mathbf{u},\mathbf{F},\mathbf{P}) = &\int_{\mathcal{B}_0} \left(\Psi(\mathbf{F}) - \mathbf{P}:\mathbf{F} - \mathbf{b}_0\cdot\mathbf{u} - \operatorname{Div}\mathbf{P}\cdot\mathbf{u}\right)\mathrm{d}V \\ &+ \int_{\partial\mathcal{B}_{0\sigma}} \mathbf{u}\cdot(\mathbf{t}_0 - \bar{\mathbf{t}}_0)\,\mathrm{d}A - \int_{\partial\mathcal{B}_{0u}} \mathbf{t}_0\cdot(\mathbf{u}-\bar{\mathbf{u}})\,\mathrm{d}A\end{aligned} \tag{2.78}$$

By requiring the three-field potential (2.78) to be stationary, the field equations and boundary conditions can be derived:

$$\begin{aligned}\operatorname{Div}\mathbf{P} + \mathbf{b}_0 &= \mathbf{0} \\ \mathbf{P} &= \frac{\partial\Psi(\mathbf{F})}{\partial\mathbf{F}} \\ \mathbf{F} &= \operatorname{Grad}\mathbf{u} + \mathbf{I} \qquad \text{in } \mathcal{B}_0 \\ \mathbf{u} &= \bar{\mathbf{u}} \qquad \text{on } \partial\mathcal{B}_{0u} \\ \mathbf{t}_0 = \mathbf{P}\mathbf{N} &= \bar{\mathbf{t}}_0 \qquad \text{on } \partial\mathcal{B}_{0\sigma}\end{aligned} \tag{2.79}$$

Here, $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}_0$ denote the prescribed boundary displacements and loads, respectively. In terms of a shell formulation, the Hu-Washizu variational principle can be employed in a similar way, which will be introduced later.

Chapter 3

Nonlinear shell theory

As the sandwich structures investigated in this work will be modeled using shell elements, the fundamental equations of nonlinear shell theory are introduced in this chapter. The shell model is based upon works from Gruttmann et al. [26, 54, 94] and will be employed by means of multiscale modeling both on the global macroscopic scale, as well as on the local scale in terms of modeling a representative volume element. A more comprehensive overview on the topic of shell structures can be found in various textbooks, e.g. [10, 93, 97].

When modeling a three-dimensional continuum as a shell, it is described by a two-dimensional reference surface and a director vector, which, in the reference configuration, is perpendicular to the reference surface. As deformations in the direction of thickness are often small, they will be neglected in terms of the so-called inextensible theory. Another common assumption is that plane cross sections remain plane, comparable to the theory of beams and plates. This leads to admissible shear deformations while requiring only C^0-continuous interpolations, contrary to the classical Kirchhoff-Love theory where C^1-continuous interpolation functions are necessary.

In order to utilize the specific geometry of a shell, the relations between deformation and strain, which were derived in the last chapter for generic continuum bodies, are adjusted and lead to the so-called shell strains in the terms of shell kinematics. With that help, stress resultants of the shell are introduced as a replacement of stresses in a generic continuum. Comparable to the case of the general continuum, the equilibrium conditions can subsequently be obtained in the shell formulation by applying the principle of virtual work.

3.1 Shell kinematics

In the following, indexes with Latin letters $i, j, \ldots$ are considered to go from 1 to 3, Greek indexes α, β from 1 to 2 and the summation convention is employed for duplicate indexes.

Starting from the Euclidean space with basis $\mathbf{e}_i$, which is used for both the reference configuration and the current configuration, a convective coordinate system ξ^i is introduced, where the coordinate lines are subject to the same deformation as the continuum body. Within the shell, which in the reference configuration occupies the shell space $\mathcal{B}_0$, a reference surface Ω_0 with boundary Γ_0 is defined. Analogously, $\mathcal{B}_t, \Omega_t$ and Γ_t denote the deformed shell space, reference surface and boundary in the current configuration at the time t. Instead of using a direct representation with a position vector $\boldsymbol{\Phi}$, each point in the undeformed shell space $\mathcal{B}_0$ can be identified by a combination of a position vector $\mathbf{X}$ to a point in the shell's reference surface and a director vector $\mathbf{D}$, orthogonal to the reference surface. In this formulation, both vectors $\mathbf{X}$ and $\mathbf{D}$ are only dependent on the convective coordinates in the shell surface, ξ^1 and ξ^2.

$$\begin{aligned} \boldsymbol{\Phi} &= \boldsymbol{\Phi}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = \Phi_1 \mathbf{e}_1 + \Phi_2 \mathbf{e}_2 + \Phi_3 \mathbf{e}_3 \\ &= \boldsymbol{\Phi}(\xi^1, \xi^2, \xi^3) = \mathbf{X}(\xi^1, \xi^2) + \xi^3 \mathbf{D}(\xi^1, \xi^2) \end{aligned} \tag{3.1}$$

In the deformed shell space of the current configuration, points $\boldsymbol{\phi}$ are identified in the same way with position vectors $\mathbf{x}$ to the deformed reference surface and the corresponding director $\mathbf{d}$:

$$\boldsymbol{\phi} = \boldsymbol{\phi}(\xi^1, \xi^2, \xi^3) = \mathbf{x}(\xi^1, \xi^2) + \xi^3 \mathbf{d}(\xi^1, \xi^2) \tag{3.2}$$

Consequently, the displacement field $\mathbf{u}$ can be introduced,

$$\mathbf{u}(\xi^1, \xi^2) = \mathbf{x}(\xi^1, \xi^2) - \mathbf{X}(\xi^1, \xi^2) \quad , \tag{3.3}$$

leading to three displacement degrees of freedom along the ξ^1, ξ^2, ξ^3 axes. The director vector $\mathbf{D}$ in the reference configuration is introduced as unit normal vector of the reference surface Ω_0, thus it has the length 1. The requirement of inextensibility is enforced by demanding the length of the director $\mathbf{d}$ in the current configuration to have the same length 1. Consequently, the shell is unable to expand in the direction of thickness.

$$|\mathbf{D}(\xi^1, \xi^2)| = |\mathbf{d}(\xi^1, \xi^2)| = 1 \tag{3.4}$$

Contrary to $\mathbf{D}$, the deformed director $\mathbf{d}$ in the current configuration is not required to be perpendicular to the deformed reference surface Ω_t. Due to this, the model permits the occurrence of transversal shear deformation, comparable to the Reissner-Mindlin plate theory. Thus, the numerical problems of the classical Kirchhoff theory [93], resulting

from the C^1 regularity requirement, do not arise in this formulation.

In the following, let the expression $(.)_{,i}$ denote the partial derivative of a quantity $(.)$ with respect to the coordinate ξ^i. In Section 2.1, the covariant bases $\mathbf{G}_i$ and $\mathbf{g}_i$ in Euclidean space were already introduced. By the means of shell theory they represent basis systems whose reference points are fixed material points in the shell. The covariant basis vectors in the reference configuration can be derived from the partial derivatives of the position vectors $\mathbf{\Phi}$ from (3.1):

$$\begin{aligned} \mathbf{G}_1 &= \frac{\partial \mathbf{\Phi}}{\partial \xi^1} = \mathbf{X}_{,1} + \xi^3 \mathbf{D}_{,1} \\ \mathbf{G}_2 &= \frac{\partial \mathbf{\Phi}}{\partial \xi^2} = \mathbf{X}_{,2} + \xi^3 \mathbf{D}_{,2} \\ \mathbf{G}_3 &= \frac{\partial \mathbf{\Phi}}{\partial \xi^3} = \mathbf{D} \end{aligned} \tag{3.5}$$

In the same way, the covariant basis vectors $\mathbf{g}_i$ in the current configuration are obtained:

$$\begin{aligned} \mathbf{g}_1 &= \frac{\partial \boldsymbol{\phi}}{\partial \xi^1} = \mathbf{x}_{,1} + \xi^3 \mathbf{d}_{,1} \\ \mathbf{g}_2 &= \frac{\partial \boldsymbol{\phi}}{\partial \xi^2} = \mathbf{x}_{,2} + \xi^3 \mathbf{d}_{,2} \\ \mathbf{g}_3 &= \frac{\partial \boldsymbol{\phi}}{\partial \xi^3} = \mathbf{d} \end{aligned} \tag{3.6}$$

The contravariant basis vectors $\mathbf{G}^j$ and $\mathbf{g}^j$ can be derived from the orthogonality conditions (2.4) in the same way as in the last chapter. With the help of these bases, the tangent vectors $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{a}_1, \mathbf{a}_2$ to the reference surface in the reference configuration and the current configuration are introduced.

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{G}_1(\xi^1, \xi^2, 0) = \mathbf{X}_{,1} \\ \mathbf{A}_2 &= \mathbf{G}_2(\xi^1, \xi^2, 0) = \mathbf{X}_{,2} \\ \mathbf{a}_1 &= \mathbf{g}_1(\xi^1, \xi^2, 0) = \mathbf{x}_{,1} \\ \mathbf{a}_2 &= \mathbf{g}_2(\xi^1, \xi^2, 0) = \mathbf{x}_{,2} \end{aligned} \tag{3.7}$$

In the reference configuration, the director vector $\mathbf{D}$ is orthogonal to the vectors $\mathbf{A}_1$ and $\mathbf{A}_2$, which are contained in the reference surface. Together with $\mathbf{A}_3 = \mathbf{D}$, they form a full local basis system, as seen in Figure 3.1. Infinitesimal area and volume elements can then be introduced in the following way, with j denoting the Jacobian in the reference surface:

$$\begin{aligned} \mathrm{d}V &= \sqrt{G}\, \mathrm{d}\xi^1 \mathrm{d}\xi^2 \mathrm{d}\xi^3 \quad \text{with} \quad \sqrt{G} = (\mathbf{G}_1 \times \mathbf{G}_2) \cdot \mathbf{G}_3 \\ \mathrm{d}A &= j\, \mathrm{d}\xi^1 \mathrm{d}\xi^2 \quad \text{with} \quad j = \sqrt{A} = |\mathbf{A}_1 \times \mathbf{A}_2| \end{aligned} \tag{3.8}$$

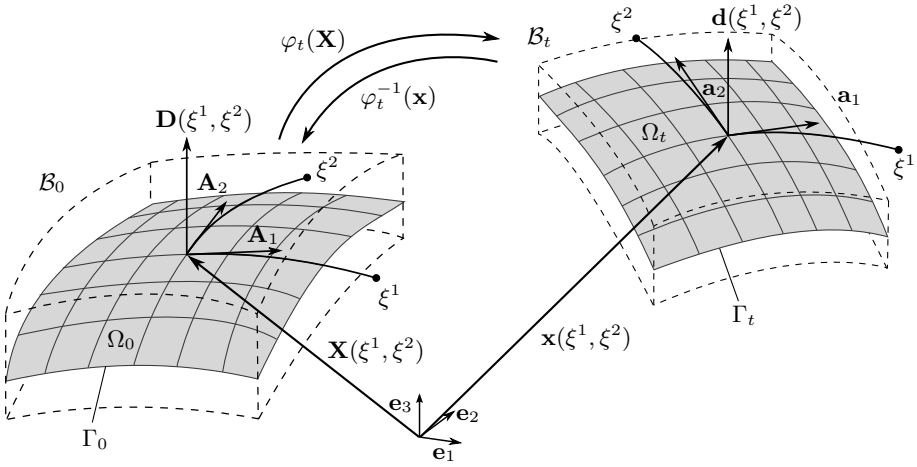


Figure 3.1: A shell in the reference configuration and the current configuration.

Using the contravariant basis vectors $\mathbf{A}^i$ and $\mathbf{a}^i$, the bases in shell space $\mathbf{G}_i$, $\mathbf{g}_i$ are correlated to the bases in the reference surface $\mathbf{A}_i$, $\mathbf{a}_i$ by defining shifter tensors $\mathbf{Z} = \mathbf{G}_i \otimes \mathbf{A}^i$ and $\bar{\mathbf{Z}} = \mathbf{g}_i \otimes \mathbf{a}^i$, so that:

$$\begin{aligned} \mathbf{G}_i &= \mathbf{Z}\mathbf{A}_i \\ \mathbf{g}_i &= \bar{\mathbf{Z}}\mathbf{a}_i \end{aligned} \tag{3.9}$$

The deformed director $\mathbf{d}$ can be obtained with help of the basis vectors in the reference surface and the undeformed director vector $\mathbf{D}$,

$$\mathbf{d} = (\mathbf{a}_i \otimes \mathbf{A}_i)\,\mathbf{D} \quad . \tag{3.10}$$

$\mathbf{R} := (\mathbf{a}_i \otimes \mathbf{A}_i)$ is an orthogonal tensor which can be expressed in several different formulations [9]. For the representation which will be used in the following, three rotational parameters $\omega_1, \omega_2, \omega_3$ with respect to the global basis $\mathbf{e}_i$ are introduced.

By defining

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \quad \omega = |\boldsymbol{\omega}|, \qquad \boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{3.11}$$

it holds $\boldsymbol{\omega} \times \mathbf{h} = \boldsymbol{\Omega}\mathbf{h}$ for any $\mathbf{h} \in \mathbb{R}^3$. The so-called Euler-Rodrigues formula can then be expressed as

$$\mathbf{R} = \mathbf{I} + \frac{\sin\omega}{\omega}\,\boldsymbol{\Omega} + \frac{1-\cos\omega}{\omega^2}\,\boldsymbol{\Omega}^2 \quad . \tag{3.12}$$

In this formulation, three rotational degrees of freedom have been introduced by means of the pseudo-rotation vector $\boldsymbol{\omega}$, describing the rotations around the ξ^1-, ξ^2- and ξ^3-axes. Combining this with the three spatial translation components of $\mathbf{u}$ in (3.3), the shell comprises six degrees of freedom. The sixth degree of freedom denotes the local drilling perpendicular to the shell surface and proves beneficial when modeling thin-walled structures with non-smooth edges or intersections, which occur in discretizations of beams with box sections or T-sections, for instance, and also when discretizing the microstructure of the sandwich panels used in this work. When dealing with smooth surfaces, the sixth degree of freedom can be forced to vanish by setting explicit boundary conditions.

3.1.1 Green-Lagrange strains and shell strains

In order to describe the deformation of a shell occurring in the transition from the reference configuration to the current configuration, suitable deformation measures must be introduced, analogously to the generic continuum. By means of the shifter tensors $\mathbf{Z}$ and $\bar{\mathbf{Z}}$ from (3.9), the deformation gradient can be expressed with respect to the tangent vectors to the reference surface. Defining $\hat{\mathbf{F}} = \mathbf{F}(\xi^3 = 0)$ and using (2.11), one obtains:

$$\begin{aligned}
\hat{\mathbf{F}} &= \mathbf{a}_i \otimes \mathbf{A}^i \\
\mathbf{a}_i &= \hat{\mathbf{F}}\mathbf{A}_i \\
\mathbf{F} &= \bar{\mathbf{Z}}\hat{\mathbf{F}}\mathbf{Z}^{-1} \\
\mathbf{g}_i &= \mathbf{F}\mathbf{G}_i = \left(\bar{\mathbf{Z}}\hat{\mathbf{F}}\mathbf{Z}^{-1}\right)\mathbf{G}_i
\end{aligned} \tag{3.13}$$

It is advantageous to introduce the Green-Lagrange strain tensor with respect to the basis system $\mathbf{A}_i$ as well, which leads with the help of $\mathbf{Z}$ to

$$\mathbf{E} = \mathbf{Z}^{-T}\hat{\mathbf{E}}\mathbf{Z}^{-1} \quad , \tag{3.14}$$

where $\hat{\mathbf{E}}$ denotes the strain tensor relating to the reference surface, which can be expressed as follows:

$$\hat{\mathbf{E}} = \hat{E}_{ij}\,\mathbf{A}^i \otimes \mathbf{A}^j = (\mathbf{E}^0 + \xi^3\mathbf{E}^1 + (\xi^3)^2\mathbf{E}^2) \tag{3.15}$$

with

$$\begin{aligned}
\mathbf{E}^0 &= E^0_{ij}\,\mathbf{A}^i \otimes \mathbf{A}^j = \varepsilon_{\alpha\beta}\,\mathbf{A}^\alpha \otimes \mathbf{A}^\beta + \frac{1}{2}\gamma_\alpha(\mathbf{A}^\alpha \otimes \mathbf{A}^3 + \mathbf{A}^3 \otimes \mathbf{A}^\alpha) \\
\mathbf{E}^1 &= E^1_{ij}\,\mathbf{A}^i \otimes \mathbf{A}^j = \kappa_{\alpha\beta}\,\mathbf{A}^\alpha \otimes \mathbf{A}^\beta \\
\mathbf{E}^2 &= E^2_{ij}\,\mathbf{A}^i \otimes \mathbf{A}^j = \rho_{\alpha\beta}\,\mathbf{A}^\alpha \otimes \mathbf{A}^\beta
\end{aligned} \tag{3.16}$$

The shell membrane strains $\varepsilon_{\alpha\beta}$, curvatures $\kappa_{\alpha\beta}$, shear strains γ_α and second-order curvatures $\rho_{\alpha\beta}$ have been introduced in (3.16). Exploiting the inextensibility condition $|\mathbf{D}| = |\mathbf{d}| = 1$, they are given by:

$$\begin{aligned}
\varepsilon_{\alpha\beta} &= \frac{1}{2}(\mathbf{x}_{,\alpha}\cdot\mathbf{x}_{,\beta} - \mathbf{X}_{,\alpha}\cdot\mathbf{X}_{,\beta}) \\
\kappa_{\alpha\beta} &= \frac{1}{2}(\mathbf{x}_{,\alpha}\cdot\mathbf{d}_{,\beta} + \mathbf{x}_{,\beta}\cdot\mathbf{d}_{,\alpha} - \mathbf{X}_{,\alpha}\cdot\mathbf{D}_{,\beta} - \mathbf{X}_{,\beta}\cdot\mathbf{D}_{,\alpha}) \\
\gamma_\alpha &= \mathbf{x}_{,\alpha}\cdot\mathbf{d} - \mathbf{X}_{,\alpha}\cdot\mathbf{D} \\
\rho_{\alpha\beta} &= \frac{1}{2}(\mathbf{d}_{,\alpha}\cdot\mathbf{d}_{,\beta} - \mathbf{D}_{,\alpha}\cdot\mathbf{D}_{,\beta})
\end{aligned} \tag{3.17}$$

In the case of thin shells, the second-order curvatures $\rho_{\alpha\beta}$ can be neglected while preserving adequate accuracy. In order to simplify the notation, the 6 components of the Green-Lagrange strain tensor $\mathbf{E}$ are arranged in a vector $\bar{\mathbf{E}}$ as follows

$$\bar{\mathbf{E}} = [E_{11}, E_{22}, E_{33}, 2E_{12}, 2E_{13}, 2E_{23}]^T \quad , \tag{3.18}$$

while the 8 components of the shell strains are arranged in the vector $\boldsymbol{\varepsilon}$ by means of

$$\boldsymbol{\varepsilon} = [\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}, \kappa_{11}, \kappa_{22}, 2\kappa_{12}, \gamma_1, \gamma_2]^T \quad . \tag{3.19}$$

Introducing the assembly matrix $\mathbf{A}$, the Green-Lagrange strain components can be related to the shell strains in the following way:

$$\begin{bmatrix} E_{11} \\ E_{22} \\ E_{33} \\ 2E_{12} \\ 2E_{13} \\ 2E_{23} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \xi^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \xi^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \xi^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \\ \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \\ \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{3.20}$$

$$\bar{\mathbf{E}} = \mathbf{A}\boldsymbol{\varepsilon}$$

In this notation, it is evident that the normal strain E_{33} always vanishes, according to the assumptions of the shell theory. It is possible to use formulations where E_{33} is completely removed from the description, for instance by using constitutive laws assuming plane stress [26].

3.2 Stress and stress resultants

The second Piola-Kirchhoff stress tensor $\mathbf{S}$, which is work conform to the Green-Lagrange strain tensor, shall also be expressed using the covariant basis vectors (3.5). The transformation is obtained in a similar way as for the Green-Lagrange strain tensor (3.14) by

$$\mathbf{S} = \mathbf{Z}\hat{\mathbf{S}}\mathbf{Z}^T \quad \text{with} \quad \hat{\mathbf{S}} = S^{ij}\,\mathbf{A}_i \otimes \mathbf{A}_j \quad . \tag{3.21}$$

The stress resultants $\mathbf{n}^\alpha, \mathbf{m}^\alpha$ are then introduced as integrals of the first Piola-Kirchhoff stress tensor over the shell thickness;

$$\begin{aligned} \mathbf{n}^{\alpha} &= \int_{h^-}^{h^+} \mathbf{P}\mathbf{G}^{\alpha}\, \bar{\mu} \mathrm{d}\xi^3 \\ \mathbf{m}^{\alpha} &= \mathbf{d} \times \int_{h^-}^{h^+} \mathbf{P}\mathbf{G}^{\alpha} \xi^3\, \bar{\mu} \mathrm{d}\xi^3 \quad , \end{aligned} \tag{3.22}$$

where h^+ and h^- denote the upper and lower boundary of the shell surface. By decomposing the stress resultants $\mathbf{n}^a$ and the director stress resultants $\mathbf{m}^{\alpha}$ into their components along $\mathbf{x}_{,\alpha}$ and $\mathbf{d}$, the description

$$\begin{aligned} \mathbf{n}^{\alpha} &= n^{\alpha\beta}\mathbf{x}_{,\beta} + q^{\alpha}\mathbf{d} + m^{\alpha\beta}\mathbf{d}_{,\beta} \\ \mathbf{m}^{\alpha} &= \mathbf{d} \times m^{\alpha\beta}\mathbf{x}_{,\beta} \end{aligned} \tag{3.23}$$

can be derived. Here, the membrane forces $n^{\alpha\beta} = n^{\beta\alpha}$, bending moments $m^{\alpha\beta} = m^{\beta\alpha}$ and shear forces q^{α} have been introduced, which are assembled in the vector of stress resultants $\boldsymbol{\sigma}$ in the following way:

$$\boldsymbol{\sigma} = \left[n^{11}, n^{22}, n^{12}, m^{11}, m^{22}, m^{12}, q^1, q^2\right]^T \tag{3.24}$$

The relation between $\mathbf{S}$ and $\boldsymbol{\sigma}$ can be expressed as follows:

$$\boldsymbol{\sigma} = \int_{h^-}^{h^+} \mathbf{A}^T \bar{\mathbf{S}}\, \bar{\mu}\, \mathrm{d}\xi^3 \tag{3.25}$$

with $\bar{\mathbf{S}} = [S_{11}, S_{22}, S_{33}, S_{12}, S_{13}, S_{23}]^T$ and $\mathbf{A}$ as in (3.20).

3.3 Equilibrium conditions

In order to obtain the equilibrium conditions in terms of the stress resultants, the variations of the shell strains $\delta\boldsymbol{\varepsilon}$ must be derived. Applying the directional derivative onto the components of $\boldsymbol{\varepsilon}$ yields with

$$\begin{aligned}
\delta\varepsilon_{\alpha\beta} &= \frac{1}{2}\left(\delta\mathbf{x}_{,\alpha}\cdot\mathbf{x}_{,\beta}+\delta\mathbf{x}_{,\beta}\cdot\mathbf{x}_{,\alpha}\right)\\
\delta\kappa_{\alpha\beta} &= \frac{1}{2}\left(\delta\mathbf{x}_{,\alpha}\cdot\mathbf{d}_{,\beta}+\delta\mathbf{x}_{,\beta}\cdot\mathbf{d}_{,\alpha}+\delta\mathbf{d}_{,\alpha}\cdot\mathbf{x}_{,\beta}+\delta\mathbf{d}_{,\beta}\cdot\mathbf{x}_{,\alpha}\right)\\
\delta\gamma_{\alpha} &= \delta\mathbf{x}_{,\alpha}\cdot\mathbf{d}+\delta\mathbf{d}\cdot\mathbf{x}_{,a}
\end{aligned} \tag{3.26}$$

the variations of the membrane strains $\delta\varepsilon_{\alpha\beta}$, the variations of the curvatures $\delta\kappa_{\alpha\beta}$ and the variations of the transverse shear strains $\delta\gamma_{\alpha}$. Analogously to (3.19), they are arranged in the vector of virtual shell strains

$$\delta\boldsymbol{\varepsilon} = \left[\delta\varepsilon_{11},\delta\varepsilon_{22},2\delta\varepsilon_{12},\delta\kappa_{11},\delta\kappa_{22},2\delta\kappa_{12},\delta\gamma_1,\delta\gamma_2\right]^T \quad . \tag{3.27}$$

The internal virtual work can then be introduced using (2.66),

$$\delta W_{\text{int}} = \int_{\mathcal{B}_0} \delta\mathbf{E} : \mathbf{S}\,\mathrm{d}V \quad , \tag{3.28}$$

by integrating over the shell body, or in terms of the stress resultants (3.24), which are already integrated over the shell thickness, as

$$\delta W_{\text{int}} = \int_{\Omega_0} \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma}\,\mathrm{d}A \quad . \tag{3.29}$$

Inserting the components of $\delta\boldsymbol{\varepsilon}$ from (3.26) and using (3.23), the representation

$$\delta W_{\text{int}} = \int_{\Omega_0} \left(\mathbf{n}^{\alpha}\cdot\delta\mathbf{x}_{,\alpha}+q^{\alpha}\mathbf{x}_{,\alpha}\cdot\delta\mathbf{d}+m^{\alpha\beta}\mathbf{x}_{,\beta}\cdot\delta\mathbf{d}_{,\alpha}\right)\mathrm{d}A \tag{3.30}$$

can be obtained. The boundary Γ_0 of the shell is separated into disjoint partial boundaries $\Gamma_{0\sigma}$ with stress boundary conditions and Γ_{0u} with displacement boundary conditions, similar to the approach for the generic continuum in the last chapter. As before, admissible variations must fulfill the kinematic boundary conditions $\delta\mathbf{x} = \delta\mathbf{u} = \delta\mathbf{d} = 0$ on Γ_{0u}. Integration by parts leads to

$$\begin{aligned}
\delta W_{\text{int}} = &-\int_{\Omega_0}\left[\frac{1}{j}(j\mathbf{n}^{\alpha})_{,\alpha}\cdot\delta\mathbf{u}+\left(\frac{1}{j}(j\mathbf{m}^{\alpha})_{,\alpha}+\mathbf{x}_{,\alpha}\times\mathbf{n}^{\alpha}\right)\cdot\delta\boldsymbol{\omega}\right]\mathrm{d}A\\
&+\int_{\Gamma_{0\sigma}}\left[j(\mathbf{n}^{\alpha}\nu_{\alpha})\cdot\delta\mathbf{u}+j(\mathbf{m}^{\alpha}\nu_{\alpha})\cdot\delta\boldsymbol{\omega}\right]\mathrm{d}s \quad ,
\end{aligned} \tag{3.31}$$

where ν_{α} describes the outer normal vector to the boundary of the two-dimensional shell

surface. Terms on Γ_{0u} vanish due to the kinematic boundary conditions. External surface loads $\bar{\mathbf{p}}$ on the reference surface Ω_0 and boundary loads $\bar{\mathbf{t}}$ on $\Gamma_{0\sigma}$ determine the external virtual work δW_{ext}:

$$\delta W_{\text{ext}} = \int_{\Omega_0} \bar{\mathbf{p}} \cdot \delta\mathbf{u} \, \mathrm{d}A + \int_{\Gamma_{0\sigma}} \bar{\mathbf{t}} \cdot \delta\mathbf{u} \, \mathrm{d}s \tag{3.32}$$

By means of that, the principle of virtual work for the shell is obtained with $\mathbf{v} := [\mathbf{u}, \boldsymbol{\omega}]^T$ as

$$g(\mathbf{v}, \delta\mathbf{v}) = \delta W_{\text{int}} - \delta W_{\text{ext}} = 0 \tag{3.33}$$

and inserting (3.31) as well as (3.32) yields the full representation

$$\begin{aligned} g(\mathbf{v}, \delta\mathbf{v}) = -\int_{\Omega_0} \left[\left(\frac{1}{j}(j\mathbf{n}^\alpha)_{,\alpha} + \bar{\mathbf{p}} \right) \cdot \delta\mathbf{u} + \left(\frac{1}{j}(j\mathbf{m}^\alpha)_{,\alpha} + \mathbf{x}_{,\alpha} \times \mathbf{n}^\alpha \right) \cdot \delta\boldsymbol{\omega} \right] \mathrm{d}A \\ + \int_{\Gamma_{0\sigma}} \left[(j\mathbf{n}^\alpha \nu_\alpha - \bar{\mathbf{t}}) \cdot \delta\mathbf{u} + j(\mathbf{m}^\alpha \nu_\alpha) \cdot \delta\boldsymbol{\omega} \right] \mathrm{d}s \quad . \end{aligned} \tag{3.34}$$

Due to arbitrarily chosen admissible variations $\delta\mathbf{u}$, $\delta\boldsymbol{\omega}$, when applying the fundamental lemma of variational calculus, the left-hand factors in the integrands must vanish, leading to the static equilibrium conditions in terms of the shell theory:

$$\begin{aligned} \frac{1}{j}(j\mathbf{n}^\alpha)_{,\alpha} + \bar{\mathbf{p}} &= 0 \quad , \\ \frac{1}{j}(j\mathbf{m}^\alpha)_{,\alpha} + \mathbf{x}_{,\alpha} \times \mathbf{n}^\alpha &= 0 \quad \text{in } \Omega_0 \end{aligned} \tag{3.35}$$

with the static boundary conditions

$$\begin{aligned} j(\mathbf{n}^\alpha \nu_\alpha) - \bar{\mathbf{t}} &= 0 \quad , \\ j(\mathbf{m}^\alpha \nu_\alpha) &= 0 \quad \text{on } \Gamma_{0\sigma} \quad . \end{aligned} \tag{3.36}$$

3.4 Mixed field formulation

In order to prevent the occurrence of locking phenomena, a mixed field formulation according to the Hu-Washizu variational principle is employed [6], which has already been introduced for a generic continuum in the last chapter.

Let

$$\boldsymbol{\varepsilon}_g := \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\mathbf{v}) = [\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}, \kappa_{11}, \kappa_{22}, 2\kappa_{12}, \gamma_1, \gamma_2]^T \quad , \tag{3.37}$$

according to (3.19), denote the geometric shell strains which depend on the displacement field. As before, $\mathbf{v} = [\mathbf{u}, \boldsymbol{\omega}]^T$ contains both translational and rotational parameters $\mathbf{u}$ and $\boldsymbol{\omega}$.

An additional field variable $\boldsymbol{\varepsilon}_p$ is introduced for the independent physical shell strains, related to the constitutive equations, which leads to the new field equation $\boldsymbol{\varepsilon}_g - \boldsymbol{\varepsilon}_p = 0$. Moreover, the independent stress resultants $\boldsymbol{\sigma}_u$ are defined, so that with the stress resultants from the constitutive law, $\boldsymbol{\sigma} = f(\boldsymbol{\varepsilon}_p)$, the further field equation $\boldsymbol{\sigma} - \boldsymbol{\sigma}_u = 0$ is given. In the case of hyperelasticity and the existence of a Helmholtz free energy function, $\boldsymbol{\sigma} = \dfrac{\partial \Psi}{\partial \boldsymbol{\varepsilon}_p}$ holds.

In total, for a shell loaded by surface loads $\bar{\mathbf{p}}$ in Ω_0 and boundary loads $\bar{\mathbf{t}}$ on $\Gamma_{0\sigma}$, the complete set of Hu-Washizu field equations follows as

$$\begin{aligned}
\frac{1}{j}(j\mathbf{n}^\alpha)_{,\alpha} + \bar{\mathbf{p}} &= 0 \quad , \\
\frac{1}{j}(j\mathbf{m}^\alpha)_{,\alpha} + \mathbf{x}_{,\alpha} \times \mathbf{n}^\alpha &= 0 \quad , \\
\boldsymbol{\varepsilon}_g - \boldsymbol{\varepsilon}_p &= 0 \quad , \\
\boldsymbol{\sigma} - \boldsymbol{\sigma}_u &= 0 \quad \text{in } \Omega_0 \quad ,
\end{aligned} \tag{3.38}$$

with static boundary conditions (3.36) on $\Gamma_{0\sigma}$, whereas the geometric boundary conditions $\mathbf{v} - \bar{\mathbf{v}} = 0$ must be fulfilled on Γ_{0u}.

Introducing $\boldsymbol{\theta} = [\mathbf{v}, \boldsymbol{\sigma}_u, \boldsymbol{\varepsilon}_p]^T$ and $\delta\boldsymbol{\theta}$ accordingly to alleviate the notation, the weak form of the stationarity condition for the field equations can be written as:

$$\begin{aligned}
g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}) = \int_{\Omega_0} \Big[& \delta\boldsymbol{\varepsilon}_p^T(\boldsymbol{\sigma} - \boldsymbol{\sigma}_u) + \delta\boldsymbol{\sigma}_u^T(\boldsymbol{\varepsilon}_g - \boldsymbol{\varepsilon}_p) - \left(\frac{1}{j} \left(j\mathbf{n}^\alpha\right)_{,\alpha} + \bar{\mathbf{p}} \right) \cdot \delta\mathbf{u} \\
& - \left(\frac{1}{j}(j\mathbf{m}^\alpha)_{,\alpha} + \mathbf{x}_{,\alpha} \times \mathbf{n}^\alpha \right) \cdot \delta\boldsymbol{\omega} \Big] \, \mathrm{d}A = 0
\end{aligned} \tag{3.39}$$

With the help of integration by parts for the last two terms and incorporating the static boundary conditions, it follows

$$g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}) = \int\limits_{\Omega_0} \left[\delta\boldsymbol{\varepsilon}_p^T (\boldsymbol{\sigma} - \boldsymbol{\sigma}_u) + \delta\boldsymbol{\sigma}_u^T (\boldsymbol{\varepsilon}_g - \boldsymbol{\varepsilon}_p) + \delta\boldsymbol{\varepsilon}_g^T \boldsymbol{\sigma}_u \right] \mathrm{d}A \\ - \int\limits_{\Omega_0} \delta\mathbf{u}^T \bar{\mathbf{p}} \, \mathrm{d}A - \int\limits_{\Gamma_{0\sigma}} \delta\mathbf{u}^T \bar{\mathbf{t}} \, \mathrm{d}s = 0 \quad , \tag{3.40}$$

where as a constraint the geometric boundary conditions must be fulfilled.

The linearized form of the variation (3.40) can be determined by evaluating the Taylor series expansion up to the linear term:

$$L[g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}), \Delta\boldsymbol{\theta}] := g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}) + \mathrm{D}\, g \cdot \Delta\boldsymbol{\theta} \tag{3.41}$$

For conservative loads, all terms corresponding to $\bar{\mathbf{p}}$ and $\bar{\mathbf{t}}$ vanish in the second variation $\mathrm{D}\, g \cdot \Delta\boldsymbol{\theta}$. This expression will be derived in an approximated form by means of the finite element method in the next chapter.

Chapter 4

Finite element formulation

In this chapter, the governing equations for the finite element method related to shell elements required for the examples in this work will be presented. A more complete overview of the theory and applications for finite element methods can be found in the literature, e.g. [4, 43, 97, 102, 103]. As in the last chapter, the formulation presented here mainly corresponds to [94].

4.1 Approximation of geometry

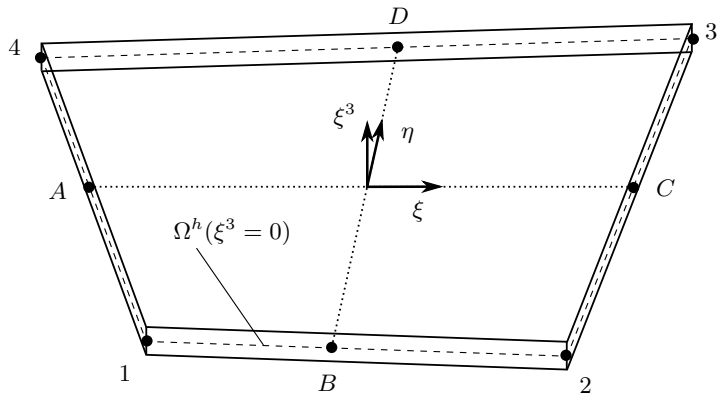


Figure 4.1: Isoparametric shell element with corner nodes $1-4$ and mid-side nodes $A-D$.

The mesh is discretized using isoparametric quadrilateral shell elements, where the in-plane coordinates $\xi := \xi^1, \eta := \xi^2$ are introduced and the unit square with coordinates $\xi, \eta \in [-1, 1]$ is mapped to the reference surface of the shell in the reference configuration and current configuration. Position vectors and director vectors of the reference surface

are thus both interpolated with bilinear functions

$$\mathbf{X}^h = \sum_{I=1}^{4} N_I \mathbf{X}_I, \quad \mathbf{D}^h = \sum_{I=1}^{4} N_I \mathbf{D}_I \tag{4.1}$$

where

$$\begin{aligned} N_1 &= \frac{1}{4}(1-\xi)(1-\eta) \quad , \\ N_2 &= \frac{1}{4}(1+\xi)(1-\eta) \quad , \\ N_3 &= \frac{1}{4}(1+\xi)(1+\eta) \quad , \\ N_4 &= \frac{1}{4}(1-\xi)(1+\eta) \quad \text{or} \\ N_I &= \frac{1}{4}(1+\xi_I\xi)(1+\eta_I\eta) \quad , \\ \mathbf{N} &= [N_1\mathbf{I}, N_2\mathbf{I}, N_3\mathbf{I}, N_4\mathbf{I}] \quad . \end{aligned} \tag{4.2}$$

The superscript $(.)^h$ in (4.1) hints at the element size in a later finite element approximation. Nodal position vectors $\mathbf{X}_I$ and the local nodal basis systems $[\mathbf{A}_{1I}, \mathbf{A}_{2I}, \mathbf{A}_{3I}]$ are determined by the mesh input, with $\mathbf{A}_{3I}$ being perpendicular to the shell reference surface Ω_0 and $\mathbf{D}_I := \mathbf{A}_{3I}$. It is possible to construct a local Cartesian basis for each element by introducing unit vectors

$$\begin{aligned} \hat{\mathbf{d}}_1 &= (\mathbf{X}_3 - \mathbf{X}_1)/|\mathbf{X}_3 - \mathbf{X}_1| \\ \hat{\mathbf{d}}_2 &= (\mathbf{X}_2 - \mathbf{X}_4)/|\mathbf{X}_2 - \mathbf{X}_4| \end{aligned} \tag{4.3}$$

and defining the local base vectors

$$\begin{aligned} \mathbf{t}_1 &= (\hat{\mathbf{d}}_1 + \hat{\mathbf{d}}_2)/|\hat{\mathbf{d}}_1 + \hat{\mathbf{d}}_2| \\ \mathbf{t}_2 &= (\hat{\mathbf{d}}_1 - \hat{\mathbf{d}}_2)/|\hat{\mathbf{d}}_1 - \hat{\mathbf{d}}_2| \\ \mathbf{t}_3 &= \mathbf{t}_1 \times \mathbf{t}_2 \quad . \end{aligned} \tag{4.4}$$

Other formulations have been proposed in [43], for instance. When using (4.4), the Jacobian matrix $\mathbf{J}$ is defined as

$$\mathbf{J} = \begin{bmatrix} \mathbf{X}^h{}_{,\xi} \cdot \mathbf{t}_1 & \mathbf{X}^h{}_{,\xi} \cdot \mathbf{t}_2 \\ \mathbf{X}^h{}_{,\eta} \cdot \mathbf{t}_1 & \mathbf{X}^h{}_{,\eta} \cdot \mathbf{t}_2 \end{bmatrix} \quad , \tag{4.5}$$

where

$$\begin{aligned} \mathbf{X}^h{}_{,\xi} &= \mathbf{G}^0_\xi + \eta \mathbf{G}^1, \quad \mathbf{G}^0_\xi = \frac{1}{4}\sum_{I=1}^{4} \xi_I \mathbf{X}_I, \\ \mathbf{X}^h{}_{,\eta} &= \mathbf{G}^0_\eta + \xi \mathbf{G}^1, \quad \mathbf{G}^0_\eta = \frac{1}{4}\sum_{I=1}^{4} \eta_I \mathbf{X}_I, \\ & \qquad\qquad \mathbf{G}^1 = \frac{1}{4}\sum_{I=1}^{4} \xi_I \eta_I \mathbf{X}_I \quad . \end{aligned} \tag{4.6}$$

$\mathbf{t}_3$ is a unit normal vector at the element center, as it is orthogonal to both $\mathbf{G}^0_\xi$ and $\mathbf{G}^0_\eta$. $\mathbf{t}_1$ and $\mathbf{t}_2$ span a tangent plane at the element center as they are orthogonal to $\mathbf{t}_3$ in turn. The tangent vectors $\mathbf{X}_{,\alpha}$ and the derivatives of the director vector $\mathbf{D}_{,\alpha}$ can be obtained with help of the inverse Jacobian matrix by means of

$$\mathbf{X}^h{}_{,\alpha} = \sum_{I=1}^{4} N_{I,\alpha}\, \mathbf{X}_I, \quad \mathbf{D}^h{}_{,\alpha} = \sum_{I=1}^{4} N_{I,\alpha}\, \mathbf{D}_I, \quad \begin{bmatrix} N_{I,1} \\ N_{I,2} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} N_{I,\xi} \\ N_{I,\eta} \end{bmatrix} \quad . \tag{4.7}$$

This way, $\mathbf{X}^h{}_{,\alpha} = \mathbf{t}_\alpha$ holds in the element center for arbitrarily warped shell elements, while in other points the vectors $\mathbf{X}^h{}_{,\alpha}$ are only approximately orthogonal. The ansatz functions for the shell in the current configuration are chosen in the same way, so that

$$\begin{aligned} \mathbf{x}^h &= \sum_{I=1}^{4} N_I \mathbf{x}_I, \quad \mathbf{d}^h = \sum_{I=1}^{4} N_I \mathbf{d}_I, \\ \mathbf{x}^h{}_{,\alpha} &= \sum_{I=1}^{4} N_{I,\alpha}\, \mathbf{x}_I, \quad \mathbf{d}^h{}_{,\alpha} = \sum_{I=1}^{4} N_{I,\alpha}\, \mathbf{d}_I, \end{aligned} \tag{4.8}$$

is obtained. In (4.8) the current position vector for each node is described by $\mathbf{x}_I = \mathbf{X}_I + \mathbf{u}_I$ and the director vectors $\mathbf{d}_I$ can be derived using the rotational transformation (3.10) with the rotational tensor (3.12), allowing for rotations $\omega_I = |\boldsymbol{\omega}_I| < 2\pi$ without singularities.

4.2 Interpolation and variation of Green-Lagrangean strains

The Green-Lagrangean strains are interpolated conforming to the interpolation of position vectors and directors in the reference and current configuration. For the transverse shear strains, the bending patch test can not be fulfilled when interpolating γ^h_α according to (3.17), see [94]. Thus, a Bathe-Dvorkin approach [17] is used for the shear strain interpolation:

$$\boldsymbol{\varepsilon}_g^h = \begin{bmatrix} \varepsilon_{11}^h \\ \varepsilon_{22}^h \\ 2\varepsilon_{12}^h \\ \kappa_{11}^h \\ \kappa_{22}^h \\ 2\kappa_{12}^h \\ \gamma_1^h \\ \gamma_2^h \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\left(\mathbf{x}^h{}_{,1}\cdot\mathbf{x}^h{}_{,1} - \mathbf{X}^h{}_{,1}\cdot\mathbf{X}^h{}_{,1}\right) \\ \frac{1}{2}\left(\mathbf{x}^h{}_{,2}\cdot\mathbf{x}^h{}_{,2} - \mathbf{X}^h{}_{,2}\cdot\mathbf{X}^h{}_{,2}\right) \\ \mathbf{x}^h{}_{,1}\cdot\mathbf{x}^h{}_{,2} - \mathbf{X}^h{}_{,1}\cdot\mathbf{X}^h{}_{,2} \\ \frac{1}{2}\left(\mathbf{x}^h{}_{,1}\cdot\mathbf{d}^h{}_{,1} - \mathbf{X}^h{}_{,1}\cdot\mathbf{D}^h{}_{,1}\right) \\ \frac{1}{2}\left(\mathbf{x}^h{}_{,2}\cdot\mathbf{d}^h{}_{,2} - \mathbf{X}^h{}_{,2}\cdot\mathbf{D}^h{}_{,2}\right) \\ \mathbf{x}^h{}_{,1}\cdot\mathbf{d}^h{}_{,2} + \mathbf{x}^h{}_{,2}\cdot\mathbf{d}^h{}_{,1} - \mathbf{X}^h{}_{,1}\cdot\mathbf{D}^h{}_{,2} - \mathbf{X}^h{}_{,2}\cdot\mathbf{D}^h{}_{,1} \\ \mathbf{J}^{-1}\begin{Bmatrix} \frac{1}{2}[(1-\eta)\gamma_\xi^B + (1+\eta)\gamma_\xi^D] \\ \frac{1}{2}[(1-\xi)\gamma_\eta^A + (1+\xi)\gamma_\eta^C] \end{Bmatrix} \end{bmatrix} , \tag{4.9}$$

where $\gamma_{\{\xi,\eta\}}^{\{A,B,C,D\}}$ denote the strains at the mid-side nodes of the shell and are given by

$$\begin{aligned} \gamma_\xi^M &= [\mathbf{x}_{,\xi}\cdot\mathbf{d} - \mathbf{X}_{,\xi}\cdot\mathbf{D}]^M \qquad (M = B, D) \\ \gamma_\eta^L &= [\mathbf{x}_{,\eta}\cdot\mathbf{d} - \mathbf{X}_{,\eta}\cdot\mathbf{D}]^L \qquad (L = A, C) \quad . \end{aligned} \tag{4.10}$$

The mid-side position vectors and directors are obtained by linear interpolation from the corresponding corner nodes:

$$\begin{aligned} \mathbf{d}^A &= \frac{1}{2}(\mathbf{d}_4 + \mathbf{d}_1), \quad \mathbf{D}^A = \frac{1}{2}(\mathbf{D}_4 + \mathbf{D}_1), \\ \mathbf{d}^B &= \frac{1}{2}(\mathbf{d}_1 + \mathbf{d}_2), \quad \mathbf{D}^B = \frac{1}{2}(\mathbf{D}_1 + \mathbf{D}_2), \\ \mathbf{d}^C &= \frac{1}{2}(\mathbf{d}_2 + \mathbf{d}_3), \quad \mathbf{D}^C = \frac{1}{2}(\mathbf{D}_2 + \mathbf{D}_3), \\ \mathbf{d}^D &= \frac{1}{2}(\mathbf{d}_3 + \mathbf{d}_4), \quad \mathbf{D}^D = \frac{1}{2}(\mathbf{D}_3 + \mathbf{D}_4), \\ \mathbf{x}^A{}_{,\eta} &= \frac{1}{2}(\mathbf{x}_4 - \mathbf{x}_1), \quad \mathbf{X}^A{}_{,\eta} = \frac{1}{2}(\mathbf{X}_4 - \mathbf{X}_1), \\ \mathbf{x}^B{}_{,\xi} &= \frac{1}{2}(\mathbf{x}_2 - \mathbf{x}_1), \quad \mathbf{X}^B{}_{,\xi} = \frac{1}{2}(\mathbf{X}_2 - \mathbf{X}_1), \\ \mathbf{x}^C{}_{,\eta} &= \frac{1}{2}(\mathbf{x}_3 - \mathbf{x}_2), \quad \mathbf{X}^C{}_{,\eta} = \frac{1}{2}(\mathbf{X}_3 - \mathbf{X}_2), \\ \mathbf{x}^D{}_{,\xi} &= \frac{1}{2}(\mathbf{x}_3 - \mathbf{x}_4), \quad \mathbf{X}^D{}_{,\xi} = \frac{1}{2}(\mathbf{X}_3 - \mathbf{X}_4) \end{aligned} \tag{4.11}$$

According to (4.9), the approximated virtual strains can be written in the following way:

$$\delta\boldsymbol{\varepsilon}_g^h = \begin{bmatrix} \delta\varepsilon_{11}^h \\ \delta\varepsilon_{22}^h \\ 2\delta\varepsilon_{12}^h \\ \delta\kappa_{11}^h \\ \delta\kappa_{22}^h \\ 2\delta\kappa_{12}^h \\ \delta\gamma_1^h \\ \delta\gamma_2^h \end{bmatrix} = \begin{bmatrix} \delta\mathbf{x}^h{}_{,1} \cdot \mathbf{x}^h{}_{,1} \\ \delta\mathbf{x}^h{}_{,2} \cdot \mathbf{x}^h{}_{,2} \\ \delta\mathbf{x}^h{}_{,1} \cdot \mathbf{x}^h{}_{,2} + \delta\mathbf{x}^h{}_{,2} \cdot \mathbf{x}^h{}_{,1} \\ \frac{1}{2}\left(\delta\mathbf{x}^h{}_{,1} \cdot \mathbf{d}^h{}_{,1} + \delta\mathbf{d}^h{}_{,1} \cdot \mathbf{x}^h{}_{,1}\right) \\ \frac{1}{2}\left(\delta\mathbf{x}^h{}_{,2} \cdot \mathbf{d}^h{}_{,2} + \delta\mathbf{d}^h{}_{,2} \cdot \mathbf{x}^h{}_{,2}\right) \\ \delta\mathbf{x}^h{}_{,1} \cdot \mathbf{d}^h{}_{,2} + \delta\mathbf{x}^h{}_{,2} \cdot \mathbf{d}^h{}_{,1} + \delta\mathbf{d}^h{}_{,1} \cdot \mathbf{x}^h{}_{,2} + \delta\mathbf{d}^h{}_{,2} \cdot \mathbf{x}^h{}_{,1} \\ \mathbf{J}^{-1} \begin{Bmatrix} \frac{1}{2}[(1-\eta)\delta\gamma_\xi^B + (1+\eta)\delta\gamma_\xi^D] \\ \frac{1}{2}[(1-\xi)\delta\gamma_\eta^A + (1+\xi)\delta\gamma_\eta^C] \end{Bmatrix} \end{bmatrix} \tag{4.12}$$

with the mid-side node virtual strains

$$\begin{aligned} \delta\gamma_\xi^M &= [\delta\mathbf{x}_{,\xi} \cdot \mathbf{d} + \mathbf{x}_{,\xi} \cdot \delta\mathbf{d}]^M \qquad (M = B, D) \\ \delta\gamma_\eta^L &= [\delta\mathbf{x}_{,\eta} \cdot \mathbf{d} + \mathbf{x}_{,\eta} \cdot \delta\mathbf{d}]^L \qquad (L = A, C) \end{aligned} \tag{4.13}$$

and the derivatives of the virtual position vectors and virtual directors given by

$$\delta\mathbf{x}^h{}_{,\alpha} = \sum_{I=1}^{4} N_{I,\alpha}\, \delta\mathbf{u}_I, \quad \delta\mathbf{d}^h{}_{,\alpha} = \sum_{I=1}^{4} N_{I,\alpha}\, \delta\mathbf{d}_I \quad . \tag{4.14}$$

In (4.14), the virtual nodal displacements $\delta\mathbf{u}_I$ and directors $\delta\mathbf{d}_I$ have been used. The derivation of $\delta\mathbf{d}_I$ is shown in full detail in [25], essentially

$$\delta\mathbf{d}_I = \delta\mathbf{w}_I \times \mathbf{d}_I \tag{4.15}$$

is obtained with

$$\delta\mathbf{w}_I = \mathbf{H}_I \delta\boldsymbol{\omega}_I, \quad \mathbf{H}_I = \mathbf{I} + \frac{1 - \cos\omega_I}{\omega_1^2}\boldsymbol{\Omega}_I + \frac{\omega_I - \sin\omega_I}{\omega_I^3}\boldsymbol{\Omega}_I^2 \quad . \tag{4.16}$$

For the shell element, a drilling stiffness is only available at nodes which are positioned on intersections. Thus, each node on a shell intersection has 6 degrees of freedom while all other nodes feature 5 degrees of freedom, as proposed in [88]. Consequently, the virtual rotation vectors $\delta\boldsymbol{\omega}_I$ must be transformed to the local coordinate system with

$$\delta\boldsymbol{\omega}_I = \mathbf{T}_{3I}\delta\boldsymbol{\beta}_I \tag{4.17}$$

where

$$\mathbf{T}_{3I} = \begin{cases} \mathbf{I}_{(3\times3)} \\ [\mathbf{a}_{1I}, \mathbf{a}_{2I}]_{(3\times2)} \end{cases} \quad , \quad \delta\boldsymbol{\beta}_I = \begin{cases} [\delta\beta_{xI}, \delta\beta_{yI}, \delta\beta_{zI}]^T & \text{on shell intersection nodes} \\ [\delta\beta_{1I}, \delta\beta_{2I}]^T & \text{else} \quad . \end{cases} \tag{4.18}$$

Here, $\delta\beta_{\alpha I}$ denote the virtual local rotations and the drilling degree of freedom is fixed with $\delta\beta_{3I} = 0$. By introducing

$$\mathbf{T}_I = \mathbf{W}_I^T \mathbf{H}_I \mathbf{T}_{3I} \quad , \tag{4.19}$$

$\delta\mathbf{d}_I = \mathbf{T}_I \delta\boldsymbol{\beta}_I$ is obtained and the approximated virtual shell strains (4.12) can be written in the following way:

$$\delta\boldsymbol{\varepsilon}_g^h = \begin{bmatrix} \delta\varepsilon_{11}^h \\ \delta\varepsilon_{22}^h \\ 2\delta\varepsilon_{12}^h \\ \delta\kappa_{11}^h \\ \delta\kappa_{22}^h \\ 2\delta\kappa_{12}^h \\ \delta\gamma_1^h \\ \delta\gamma_2^h \end{bmatrix} = \sum_{I=1}^{4} \begin{bmatrix} N_{I,1}\,\mathbf{x}^T{}_{,1} & 0 \\ N_{I,2}\,\mathbf{x}^T{}_{,2} & 0 \\ N_{I,1}\,\mathbf{x}^T{}_{,2} + N_{I,2}\,\mathbf{x}^T{}_{,1} & 0 \\ N_{I,1}\,\mathbf{d}^T{}_{,1} & N_{I,1}\,\mathbf{b}_{I1}^T \\ N_{I,2}\,\mathbf{d}^T{}_{,2} & N_{I,2}\,\mathbf{b}_{I2}^T \\ N_{I,1}\,\mathbf{d}^T{}_{,2} + N_{I,2}\,\mathbf{d}^T{}_{,1} & N_{I,1}\,\mathbf{b}_{I2}^T + N_{I,2}\,\mathbf{b}_{I1}^T \\ \mathbf{J}^{-1} \begin{Bmatrix} N_{I,\xi}\,\mathbf{d}_M^T \\ N_{I,\eta}\,\mathbf{d}_L^T \end{Bmatrix} & \mathbf{J}^{-1} \begin{Bmatrix} N_{I,\xi}\,\xi_I \mathbf{b}_M^T \\ N_{I,\eta}\,\eta_I \mathbf{b}_L^T \end{Bmatrix} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}_I \\ \delta\boldsymbol{\beta}_I \end{bmatrix} = \sum_{I=1}^{4} \mathbf{B}_I \delta\mathbf{v}_I \tag{4.20}$$

with

$$\mathbf{b}_{I\alpha} = \mathbf{T}_I^T \mathbf{x}_{,\alpha} \,, \quad \mathbf{b}_M = \mathbf{T}_I^T \mathbf{x}^M{}_{,\xi} \,, \quad \mathbf{b}_L = \mathbf{T}_I^T \mathbf{x}^L{}_{,\eta} \tag{4.21}$$

In (4.20), the discretization superscript $(.)^h$ has been partially omitted to alleviate the notation. The mid-side nodes $A-D$ and corner nodes $1-4$ relate to each other according to Figure 4.1, with

$$(I, M, L) = (1, B, A); (2, B, C); (3, D, C); (4, D, A) \quad . \tag{4.22}$$

4.3 Second variation of the functional

Starting from the Hu-Washizu functional (3.40) with conservative loads $\bar{\mathbf{p}}$ and $\bar{\mathbf{t}}$, the second variation of the functional reads

$$\mathrm{D}\, g \cdot \Delta\boldsymbol{\theta}^h = \int_{\Omega_0} \left[\delta\boldsymbol{\varepsilon}_p^{hT} (\boldsymbol{\mathcal{D}} \Delta\boldsymbol{\varepsilon}_p^h - \Delta\boldsymbol{\sigma}_u^h) + \delta\boldsymbol{\sigma}_u^{hT} (\Delta\boldsymbol{\varepsilon}_g^h - \Delta\boldsymbol{\varepsilon}_p^h) + \delta\boldsymbol{\varepsilon}_g^{hT} \Delta\boldsymbol{\sigma}_u^h + \Delta\delta\boldsymbol{\varepsilon}_g^{hT} \boldsymbol{\sigma}_u^h \right] \mathrm{d}A \tag{4.23}$$

with $\boldsymbol{\mathcal{D}} := \dfrac{\partial \boldsymbol{\sigma}^h}{\partial \boldsymbol{\varepsilon}_p^h}$ and for the hyperelastic case $\boldsymbol{\mathcal{D}} = \dfrac{\partial^2 \Psi}{(\partial \boldsymbol{\varepsilon}_p^h)^2}$.

In the following, the remaining quantities used in (4.23) will be introduced.

The linearized geometric shell strains $\Delta\boldsymbol{\varepsilon}_g^h$ are given by (4.12) when replacing the operator δ by Δ, while the linearized virtual shell strains $\Delta\delta\boldsymbol{\varepsilon}_g^h$ are obtained as

$$\Delta\delta\boldsymbol{\varepsilon}_g^h = \begin{bmatrix} \Delta\delta\varepsilon_{11}^h \\ \Delta\delta\varepsilon_{22}^h \\ 2\Delta\delta\varepsilon_{12}^h \\ \Delta\delta\kappa_{11}^h \\ \Delta\delta\kappa_{22}^h \\ 2\Delta\delta\kappa_{12}^h \\ \\ \Delta\delta\gamma_1^h \\ \Delta\delta\gamma_2^h \end{bmatrix} = \begin{bmatrix} \delta\mathbf{x}^h{}_{,1} \cdot \Delta\mathbf{x}^h{}_{,1} \\ \delta\mathbf{x}^h{}_{,2} \cdot \Delta\mathbf{x}^h{}_{,2} \\ \delta\mathbf{x}^h{}_{,1} \cdot \Delta\mathbf{x}^h{}_{,2} + \delta\mathbf{x}^h{}_{,2} \cdot \Delta\mathbf{x}^h{}_{,1} \\ \frac{1}{2}\left(\delta\mathbf{x}^h{}_{,1} \cdot \Delta\mathbf{d}^h{}_{,1} + \delta\mathbf{d}^h{}_{,1} \cdot \Delta\mathbf{X}^h{}_{,1} + \mathbf{x}^h{}_{,1} \cdot \Delta\delta\mathbf{d}^h{}_{,1} \right) \\ \frac{1}{2}\left(\delta\mathbf{x}^h{}_{,2} \cdot \Delta\mathbf{d}^h{}_{,2} + \delta\mathbf{d}^h{}_{,2} \cdot \Delta\mathbf{X}^h{}_{,2} + \mathbf{x}^h{}_{,2} \cdot \Delta\delta\mathbf{d}^h{}_{,2} \right) \\ \delta\mathbf{x}^h{}_{,1} \cdot \Delta\mathbf{d}^h{}_{,2} + \delta\mathbf{x}^h{}_{,2} \cdot \Delta\mathbf{d}^h{}_{,1} + \delta\mathbf{d}^h{}_{,1} \cdot \Delta\mathbf{x}^h{}_{,2} + \delta\mathbf{d}^h{}_{,2} \cdot \Delta\mathbf{x}^h{}_{,1} + \\ \quad + \mathbf{x}^h{}_{,1} \cdot \Delta\delta\mathbf{d}^h{}_{,2} + \mathbf{x}^h{}_{,2} \cdot \Delta\delta\mathbf{d}^h{}_{,1} \\ \mathbf{J}^{-1} \begin{Bmatrix} \frac{1}{2}[(1-\eta)\Delta\delta\gamma_\xi^B + (1+\eta)\Delta\delta\gamma_\xi^D] \\ \frac{1}{2}[(1-\xi)\Delta\delta\gamma_\eta^A + (1+\xi)\Delta\delta\gamma_\eta^C] \end{Bmatrix} \end{bmatrix} \tag{4.24}$$

using

$$\begin{aligned} \Delta\delta\gamma_\xi^M &= [\delta\mathbf{x}_{,\xi} \cdot \Delta\mathbf{d} + \Delta\mathbf{x}_{,\xi} \cdot \delta\mathbf{d} + \mathbf{x}_{,\xi} \cdot \Delta\delta\mathbf{d}]^M \qquad (M = B, D) \\ \Delta\delta\gamma_\eta^L &= [\delta\mathbf{x}_{,\eta} \cdot \Delta\mathbf{d} + \Delta\mathbf{x}_{,\eta} \cdot \delta\mathbf{d} + \mathbf{x}_{,\eta} \cdot \Delta\delta\mathbf{d}]^L \qquad (L = A, C) \quad . \end{aligned} \tag{4.25}$$

Deriving the second variation of the current orthogonal basis system using the Euler-

Rodrigues formula for a Taylor series expansion, as shown in [25], yields with some additional calculations [94] the finite element formulation of the linearized virtual shell membrane strains $\Delta\delta\varepsilon^h_{\alpha\beta}$ and curvatures $\Delta\delta\kappa^h_{\alpha\beta}$;

$$
\begin{aligned}
\Delta\delta\varepsilon^h_{\alpha\beta} &= \sum_{I=1}^{4}\sum_{K=1}^{4}\frac{1}{2}\left(N_{I,\alpha}\,N_{K,\beta}+N_{I,\beta}+N_{K,\alpha}\right)\delta\mathbf{u}_I\cdot\Delta\mathbf{u}_K \\
\Delta\delta\kappa^h_{\alpha\beta} &= \sum_{I=1}^{4}\sum_{K=1}^{4}\frac{1}{2}\{(N_{I,\alpha}\,N_{K,\beta}+N_{I,\beta}+N_{K,\alpha}\,)\,\delta\mathbf{u}_I\cdot\Delta\mathbf{d}_K \\
&\quad + (N_{I,\alpha}\,N_{K,\beta}+N_{I,\beta}\,N_{K,\alpha}\,)\,\delta\mathbf{d}_I\cdot\Delta\mathbf{u}_K \\
&\quad + \delta_{IK}\,[\delta\mathbf{w}_I\cdot(N_{I,\alpha}\,\mathbf{M}_I(\mathbf{x}_{,\beta}\,)+N_{I,\beta}\,\mathbf{M}_I(\mathbf{x}_{,\alpha}\,))\Delta\mathbf{w}_K]\}
\end{aligned}
\tag{4.26}
$$

where δ_{IK} denotes the Kronecker delta and $\mathbf{M}_I$ is given in [94]. The term $\Delta\delta\boldsymbol{\varepsilon}_g^{hT}\boldsymbol{\sigma}_u^h$ in (4.23) can subsequently be evaluated with the independent stress resultants $\boldsymbol{\sigma}_u^h=[n^{11},n^{22},n^{12},m^{11},m^{22},m^{12},q^1,q^2]^T$,

$$
\begin{aligned}
\Delta\delta\boldsymbol{\varepsilon}_g^{hT}\boldsymbol{\sigma}_u^h &= \sum_{I=1}^{4}\sum_{K=1}^{4}\delta\mathbf{v}_I^T\mathbf{k}_{\sigma IK}\Delta\mathbf{v}_K \\
&= \sum_{I=1}^{4}\sum_{K=1}^{4}\begin{bmatrix}\delta\mathbf{u}_I\,, & \delta\boldsymbol{\beta}_I\end{bmatrix}\begin{bmatrix}\hat{n}_{IK}\mathbf{I} & (\hat{m}_{IK}+\hat{q}^{uw}_{IK})\mathbf{T}_K\\ (\hat{m}_{IK}+\hat{q}^{wu}_{IK})\mathbf{T}_I^T & \delta_{IK}\hat{\mathbf{M}}_I(\mathbf{h}_I)\end{bmatrix}\begin{bmatrix}\Delta\mathbf{u}_K\\ \Delta\boldsymbol{\beta}_K\end{bmatrix}
\end{aligned}
\tag{4.27}
$$

where the entries of the matrix $\mathbf{k}_{\sigma IK}$ are given by

$$
\begin{aligned}
\hat{n}_{IK} &= n^{11}N_{I,1}\,N_{K,1}+n^{22}N_{I,2}\,N_{K,2}+n^{12}(N_{I,1}\,N_{K,2}+N_{I,2}\,N_{K,1}\,)\\
\hat{m}_{IK} &= m^{11}N_{I,1}\,N_{K,1}+m^{22}N_{I,2}\,N_{K,2}+m^{12}(N_{I,1}\,N_{K,2}+N_{I,2}\,N_{K,1}\,)\\
\hat{q}^{uw}_{IK} &= \frac{1}{2}(q^{\xi}N_{I,\xi}\,f^1_{IK}+q^{\eta}N_{I,\eta}\,f^2_{IK})\\
\hat{q}^{wu}_{IK} &= \frac{1}{2}(q^{\xi}N_{K,\xi}\,f^1_{IK}+q^{\eta}N_{K,\eta}\,f^2_{IK})\\
\hat{\mathbf{M}}_I &= \mathbf{T}_{3I}^T\mathbf{H}_I^T\mathbf{M}_I(\mathbf{h}_I)\mathbf{H}_I\mathbf{T}_{3I}
\end{aligned}
\tag{4.28}
$$

$$\begin{aligned}\mathbf{h}_I &= m^{11} N_{I,1}\, \mathbf{x}^h{}_{,1} + m^{22} N_{I,2}\, \mathbf{x}^h{}_{,2} + m^{12}(N_{I,2}\, \mathbf{x}^h{}_{,1} + N_{I,1}\, \mathbf{x}^h{}_{,2}) \\ &\quad + q^\xi N_{I,\xi}\, \xi_I \mathbf{x}^M{}_{,\xi} + q^\eta N_{I,\eta}\, \eta_I \mathbf{x}^L{}_{,\eta} \\ f^1_{IK} &= \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad f^2_{IK} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} q^\xi \\ q^\eta \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} q^1 \\ q^2 \end{bmatrix} \quad . \end{aligned} \tag{4.29}$$

4.4 Interpolation of stress resultants and shell strains

The independent stress resultants $\boldsymbol{\sigma}_u$ are approximated according to [94] with $\boldsymbol{\sigma}^h_u$, in order to ensure fulfillment of the patch test as well as stability requirements. In some of the following matrices the subscript u from $\boldsymbol{\sigma}_u$ is omitted to simplify the notation.

$$\boldsymbol{\sigma}^h_u = \mathbf{N}_\sigma \hat{\boldsymbol{\sigma}}_u, \quad \mathbf{N}_\sigma = \left[\mathbf{I}_{(8\times 8)}, \tilde{\mathbf{N}}_\sigma \right]_{(8\times 14)} \tag{4.30}$$

with

$$\begin{aligned} \tilde{\mathbf{N}}_\sigma &= \begin{bmatrix} \mathbf{N}^m_\sigma & 0 & 0 \\ 0 & \mathbf{N}^b_\sigma & 0 \\ 0 & 0 & \mathbf{N}^s_\sigma \end{bmatrix} \\ \mathbf{N}^m_\sigma = \mathbf{N}^b_\sigma &= \begin{bmatrix} J^0_{11} J^0_{11} (\eta - \bar{\eta}) & J^0_{21} J^0_{21} (\xi - \bar{\xi}) \\ J^0_{12} J^0_{12} (\eta - \bar{\eta}) & J^0_{22} J^0_{22} (\xi - \bar{\xi}) \\ J^0_{11} J^0_{12} (\eta - \bar{\eta}) & J^0_{21} J^0_{22} (\xi - \bar{\xi}) \end{bmatrix} \\ \mathbf{N}^s_\sigma &= \begin{bmatrix} J^0_{11} (\eta - \bar{\eta}) & J^0_{21} (\xi - \bar{\xi}) \\ J^0_{12} (\eta - \bar{\eta}) & J^0_{22} (\xi - \bar{\xi}) \end{bmatrix} \end{aligned} \tag{4.31}$$

The coefficients $J^0_{\alpha\beta}$ are the components of the Jacobian matrix $\mathbf{J}$ from (4.5), evaluated

at the element center $\xi, \eta = 0$. With the element surface area $A_e = \int_{\Omega_e} \mathrm{d}A$,

$$\bar{\xi} = \frac{1}{A_e} \int_{\Omega_e} \xi \, \mathrm{d}A, \quad \bar{\eta} = \frac{1}{A_e} \int_{\Omega_e} \eta \, \mathrm{d}A \tag{4.32}$$

are given as constants. In this formulation, the vector $\hat{\boldsymbol{\sigma}}_u$ comprises 8 parameters for the constant and 6 parameters for the varying part of the stress field, leading to partially decoupled matrices.

In a similar way, the independent shell strains $\boldsymbol{\varepsilon}_p$ are approximated using the same shape functions. Again, the subscript p from $\boldsymbol{\varepsilon}_p$ is omitted in the according matrices.

$$\boldsymbol{\varepsilon}_p^h = \mathbf{N}_\varepsilon \, \hat{\boldsymbol{\varepsilon}}_p, \quad \mathbf{N}_\varepsilon = \left[\mathbf{I}_{(8\times 8)}, \tilde{\mathbf{N}}_\varepsilon \right]_{(8\times 14)} \tag{4.33}$$

with

$$\begin{aligned}
\tilde{\mathbf{N}}_\varepsilon &= \begin{bmatrix} \mathbf{N}_\varepsilon^m & 0 & 0 \\ 0 & \mathbf{N}_\varepsilon^b & 0 \\ 0 & 0 & \mathbf{N}_\varepsilon^s \end{bmatrix} \\
\mathbf{N}_\varepsilon^m = \mathbf{N}_\varepsilon^b &= \begin{bmatrix} J_{11}^0 J_{11}^0 (\eta - \bar{\eta}) & J_{21}^0 J_{21}^0 (\xi - \bar{\xi}) \\ J_{12}^0 J_{12}^0 (\eta - \bar{\eta}) & J_{22}^0 J_{22}^0 (\xi - \bar{\xi}) \\ 2 J_{11}^0 J_{12}^0 (\eta - \bar{\eta}) & 2 J_{21}^0 J_{22}^0 (\xi - \bar{\xi}) \end{bmatrix} \\
\mathbf{N}_\varepsilon^s = \mathbf{N}_\sigma^s &= \begin{bmatrix} J_{11}^0 (\eta - \bar{\eta}) & J_{21}^0 (\xi - \bar{\xi}) \\ J_{12}^0 (\eta - \bar{\eta}) & J_{22}^0 (\xi - \bar{\xi}) \end{bmatrix} \quad .
\end{aligned} \tag{4.34}$$

4.5 Linearized variational functional

With the former interpolations for the displacement, stress and strain fields, the linearized variational functional is obtained as

$$
\begin{aligned}
L[g(\boldsymbol{\theta}^h, \delta\boldsymbol{\theta}^h), \Delta\boldsymbol{\theta}^h] :=& g(\boldsymbol{\theta}^h, \delta\boldsymbol{\theta}^h) + \mathrm{D}\, g \cdot \Delta\boldsymbol{\theta}^h \\
=& \sum_{e=1}^{numel} \begin{bmatrix} \delta\mathbf{v} \\ \delta\hat{\boldsymbol{\varepsilon}}_p \\ \delta\hat{\boldsymbol{\sigma}}_u \end{bmatrix}_e^T \left\{ \begin{bmatrix} \mathbf{k}_g & 0 & \mathbf{G}^T \\ 0 & \mathbf{H} & -\mathbf{F} \\ \mathbf{G} & -\mathbf{F}^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{V} \\ \Delta\hat{\boldsymbol{\varepsilon}}_p \\ \Delta\hat{\boldsymbol{\sigma}}_u \end{bmatrix} + \begin{bmatrix} \mathbf{f}^i - \mathbf{f}^a \\ \mathbf{f}^e \\ \mathbf{f}^s \end{bmatrix} \right\}_e
\end{aligned}
\tag{4.35}
$$

with the submatrices

$$
\begin{aligned}
\mathbf{k}_g &= \int_{\Omega_e} \mathbf{k}_\sigma \,\mathrm{d}A & \mathbf{H} &= \int_{\Omega_e} \mathbf{N}_\varepsilon^T \boldsymbol{\mathcal{D}} \mathbf{N}_\varepsilon \,\mathrm{d}A \\
\mathbf{F} &= \int_{\Omega_e} \mathbf{N}_\varepsilon^T \mathbf{N}_\sigma \,\mathrm{d}A & \mathbf{G} &= \int_{\Omega_e} \mathbf{N}_\sigma^T \mathbf{B} \,\mathrm{d}A
\end{aligned}
\tag{4.36}
$$

and vectors

$$
\begin{aligned}
\mathbf{f}^i &= \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma}_u^h \,\mathrm{d}A = \mathbf{G}^T \hat{\boldsymbol{\sigma}}_u \\
\mathbf{f}^e &= \int_{\Omega_e} \mathbf{N}_\varepsilon^T \boldsymbol{\sigma}^h \,\mathrm{d}A - \mathbf{F}\hat{\boldsymbol{\sigma}}_u \\
\mathbf{f}^s &= \int_{\Omega_e} \mathbf{N}_\sigma^T \boldsymbol{\varepsilon}_g^h \,\mathrm{d}A - \mathbf{F}^T \hat{\boldsymbol{\varepsilon}}_p \quad .
\end{aligned}
\tag{4.37}
$$

Here,

$$
\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4] \tag{4.38}
$$

from (4.20) and $\mathbf{k}_\sigma$ from (4.27) are used. The integrals in (4.32), (4.36) and (4.37) are evaluated numerically employing a Gauss integration scheme. The derivation of stress resultants $\boldsymbol{\sigma}^h$ and linearized stress resultants $\boldsymbol{\mathcal{D}}$ is described in [94] and requires fulfillment of a plane stress condition in each integration point. It can be shown that $\mathbf{F}$ comprises entries only on the main diagonal and the first secondary diagonal, and that in the case of linear elasticity $\mathbf{f}^e$ vanishes identically.

The stationarity condition $L[g(\boldsymbol{\theta}^h, \delta\boldsymbol{\theta}^h), \Delta\boldsymbol{\theta}^h]$ yields with variations $\delta\boldsymbol{\theta}^h \neq 0$ the equation

$$\begin{bmatrix} \mathbf{k}_g & 0 & \mathbf{G}^T \\ 0 & \mathbf{H} & -\mathbf{F} \\ \mathbf{G} & -\mathbf{F}^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v} \\ \Delta\hat{\boldsymbol{\varepsilon}}_p \\ \Delta\hat{\boldsymbol{\sigma}}_u \end{bmatrix} + \begin{bmatrix} \mathbf{f}^i - \mathbf{f}^a \\ \mathbf{f}^e \\ \mathbf{f}^s \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ 0 \\ 0 \end{bmatrix} \quad . \tag{4.39}$$

As the matrix $\mathbf{F}$ can be inverted easily, the parameters $\Delta\hat{\boldsymbol{\varepsilon}}_p$ and $\Delta\hat{\boldsymbol{\sigma}}_u$ can be eliminated with

$$\begin{aligned} \Delta\hat{\boldsymbol{\varepsilon}}_p &= \mathbf{F}^{-T}(\mathbf{G}\Delta\mathbf{v} + \mathbf{f}^s) \\ \Delta\hat{\boldsymbol{\sigma}}_u &= \mathbf{F}^{-1}(\mathbf{H}\Delta\hat{\boldsymbol{\varepsilon}}_p + \mathbf{f}^e) \quad . \end{aligned} \tag{4.40}$$

Inserting these into (4.39), the tangential element stiffness matrix $\mathbf{k}_T^e$ and the element residuum vector $\hat{\mathbf{f}}$ are obtained and it follows:

$$L[g(\boldsymbol{\theta}^h, \delta\boldsymbol{\theta}^h), \Delta\boldsymbol{\theta}^h] = \sum_{e=1}^{numel} \delta\mathbf{v}^T(\mathbf{k}_T^e\Delta\mathbf{v} + \hat{\mathbf{f}}) = 0 \tag{4.41}$$

with

$$\begin{aligned} \mathbf{k}_T^e &= \mathbf{G}^T\hat{\mathbf{H}}\mathbf{G} + \mathbf{k}_g \\ \hat{\mathbf{f}} &= \mathbf{G}^T(\hat{\boldsymbol{\sigma}}_u + \hat{\mathbf{H}}\mathbf{f}^s + \mathbf{F}^{-1}\mathbf{f}^e) - \mathbf{f}^a \\ \hat{\mathbf{H}} &= \mathbf{F}^{-1}\mathbf{H}\mathbf{F}^{-T} \quad . \end{aligned} \tag{4.42}$$

Using the standard assembly operator $\mathbf{A}$, the global tangential stiffness matrix and residuum vector can be derived:

$$\begin{aligned} \mathbf{K}_T &= \mathop{\mathbf{A}}_{e=1}^{numel} \mathbf{k}_T^e \\ \hat{\mathbf{F}} &= \mathop{\mathbf{A}}_{e=1}^{numel} \hat{\mathbf{f}} \end{aligned} \tag{4.43}$$

By employing a Newton type iteration scheme, the global equation system can be solved, where $\Delta\mathbf{V} = -\mathbf{K}_T^{-1}\hat{\mathbf{F}}$ yields the increment of spatial displacements $\Delta\mathbf{u}_I$ and rotations $\Delta\boldsymbol{\beta}_I$ at each node in each iteration step. Depending on whether a shell intersection is present at a specific node, transformation (4.18) is applied to yield either 2 or 3 rotational degrees of freedom. While the spatial displacements and rotations are updated on the global system level just as in a standard displacement method, the stress and strain increments are updated on element level corresponding to (4.40).

Chapter 5

The coupled global-local shell model

5.1 Extension to the local model

In this work, coupled global-local models of sandwich shells are considered. The global-local model will be derived in this chapter, it is based upon the earlier work [27], where a two-scale shell model was obtained using 8- or 27-node brick and solid shell elements for the description of a layered microstructure.

Contrary to this approach and due to the special microstructure of the sandwich panels modeled in this work, it is expedient to use shell elements for the discretization of the local problem as well as the global problem. Consequently, the formulations of the principle of virtual work (3.40) and its linearization (4.41) for a classical shell model must be extended to additionally incorporate the description of the local model. As the same shell discretization is used on both scales, including the usage of the Hu-Washizu mixed field formulation, the derivation of the governing equations will not be fully repeated here. Instead, only the local notation of relevant measures will be introduced, as well as presenting the coupled global-local formulation.

The local shell model will be implemented by means of a constitutive law for the global shell. Thus, one local boundary value problem is solved in each integration point i of every element e in the global shell model. It is assumed that an arbitrary fixed global element has been chosen and thus the index e can be omitted to alleviate the notation. The local shell reference surface in an integration point i of the considered element is introduced as Ω_i with boundaries Γ_{iu} and $\Gamma_{i\sigma}$ for displacement and stress boundary conditions, respectively. It is noted that the local shell surfaces $\Omega_i\,(i \geq 1)$ and their boundaries do not spatially correlate to the global shell surface Ω_0 and thus all measures such as displacements, stresses or strains have to be introduced completely independent from the global model, except when explicitly stated otherwise.

The stress resultants on the local scale $\mathbf{n}^\alpha$ and $\mathbf{m}^\alpha$ are obtained by integration of the first Piola-Kirchhoff stress tensor $\mathbf{P}$ over the local shell thickness $\xi^3 \in [h_l^-, h_l^+]$:

$$\begin{aligned} \mathbf{n}^\alpha &= \int_{h_l^-}^{h_l^+} \mathbf{P}\mathbf{G}^\alpha \, \bar{\mu} \, \mathrm{d}\xi^3 \\ \mathbf{m}^\alpha &= \mathbf{d} \times \int_{h_l^-}^{h_l^+} \mathbf{P}\mathbf{G}^\alpha \xi^3 \, \bar{\mu} \, \mathrm{d}\xi^3 \end{aligned} \tag{5.1}$$

In the local model, no external surface loads or boundary loads on Ω_i or $\Gamma_{i\sigma}$ are present. Hence, $\mathbf{p}$ or $\mathbf{t}$ do not occur and using the same steps as in the global shell model, the Hu-Washizu field equations and static boundary conditions follow:

$$\begin{aligned} \frac{1}{j}(j\mathbf{n}^\alpha)_{,\alpha} &= 0 \quad , \\ \frac{1}{j}(j\mathbf{m}^\alpha)_{,\alpha} + \mathbf{x}_{,\alpha} \times \mathbf{n}^\alpha &= 0 \quad , \\ \boldsymbol{\varepsilon}_g - \boldsymbol{\varepsilon}_p &= 0 \quad , \\ \boldsymbol{\sigma} - \boldsymbol{\sigma}_u &= 0 \quad \text{in } \Omega_i \\ j(\mathbf{n}^\alpha \nu_\alpha) &= 0 \quad , \\ j(\mathbf{m}^\alpha \nu_\alpha) &= 0 \quad \text{on } \Gamma_{i\sigma} \quad . \end{aligned} \tag{5.2}$$

The geometric boundary conditions $\mathbf{v} - \tilde{\mathbf{v}} = 0$ must be met on Γ_{iu}, where in $\tilde{\mathbf{v}}$ the global-local coupling will be introduced later, as the local boundary displacements are determined by shell strains in the integration points of the global model.

To take (5.2) into account and derive the coupled global-local formulation, the principle of virtual work (3.40) of the global shell is extended. In the following, global quantities with respect to Ω_0 will be denoted with a bar, such as $\bar{\mathbf{v}} = [\bar{\mathbf{u}}, \bar{\boldsymbol{\omega}}]$, $\bar{\boldsymbol{\sigma}}_u$, and $\bar{\boldsymbol{\varepsilon}}_p$, whereas $\mathbf{v} = [\mathbf{u}, \boldsymbol{\omega}]$, $\boldsymbol{\sigma}_u$, and $\boldsymbol{\varepsilon}_p$ specify local quantities in Ω_i, $i \geq 1$. The test function $\delta\boldsymbol{\theta}$ now comprises variations both on the global and the local level:

$$\delta\boldsymbol{\theta} = [\delta\bar{\mathbf{v}}, \delta\bar{\boldsymbol{\sigma}}_u, \delta\bar{\boldsymbol{\varepsilon}}_p, \delta\mathbf{v}, \delta\boldsymbol{\sigma}_u, \delta\boldsymbol{\varepsilon}_p]^T \tag{5.3}$$

The local contributions are added to the the weak form of equilibrium, so that for a given

discretization of the global model with *numel* shell elements and *ngp* Gauss points each,

$$\begin{aligned} g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}) = & \int_{\Omega_0} \left[\delta\bar{\boldsymbol{\varepsilon}}_p^T (\bar{\boldsymbol{\sigma}} - \bar{\boldsymbol{\sigma}}_u) + \delta\bar{\boldsymbol{\sigma}}_u^T (\bar{\boldsymbol{\varepsilon}}_g - \bar{\boldsymbol{\varepsilon}}_p) + \delta\bar{\boldsymbol{\varepsilon}}_g^T \bar{\boldsymbol{\sigma}}_u \right] \mathrm{d}A \\ & - \int_{\Omega_0} \delta\bar{\mathbf{u}}^T \bar{\mathbf{p}} \, \mathrm{d}A - \int_{\Gamma_{0\sigma}} \delta\bar{\mathbf{u}}^T \bar{\mathbf{t}} \, \mathrm{d}s \\ & + \sum_{e=1}^{numel} \sum_{i=1}^{ngp} \frac{1}{A_i} \int_{\Omega_i} \left[\delta\boldsymbol{\varepsilon}_p^T (\boldsymbol{\sigma} - \boldsymbol{\sigma}_u) + \delta\boldsymbol{\sigma}_u^T (\boldsymbol{\varepsilon}_g - \boldsymbol{\varepsilon}_p) + \delta\boldsymbol{\varepsilon}_g^T \boldsymbol{\sigma}_u \right] \mathrm{d}A \end{aligned} \tag{5.4}$$

is obtained. Here, A_i represents the RVE shell surface area $A_i = \int_{\Omega_i} \mathrm{d}A$. The linearized form of variation (5.4) is derived, as shown before in Chapter 4, by evaluating the Taylor series expansion up to the linear term:

$$L[g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}), \Delta\boldsymbol{\theta}] := g(\boldsymbol{\theta}, \delta\boldsymbol{\theta}) + \mathrm{D}\, g \cdot \Delta\boldsymbol{\theta} \tag{5.5}$$

5.2 Local assembly of the coupled system

In the next step, the local geometry is approximated with a finite element mesh in the same way as the global structure. The local system describes the microstructure of the global shell by means of a representative volume element, whose exact geometry depends on the depicted composite material. Using the same isoparametric quadrilateral shell elements as in the global system, a tangential stiffness matrix $\mathbf{K}_i^L$ as well as a residuum vector $\hat{\mathbf{F}}_i$ are obtained for each integration point i in a global shell element. The local boundary value problem in the linearized weak form is assembled in a standard way,

$$\delta\mathbf{V}_i^T (\mathbf{K}_i^L \Delta\mathbf{V}_i + \hat{\mathbf{F}}_i^L) = \frac{1}{A_i} \sum_{e=1}^{n} \delta\mathbf{v}_e^T (\mathbf{k}_e^L \Delta\mathbf{v}_e + \hat{\mathbf{f}}_e^L) \tag{5.6}$$

where e and n represent element numbers and count in the local discretization of the RVE and the element tangential stiffness matrices $\mathbf{k}_e^L$ and residuum vectors $\hat{\mathbf{f}}_e^L$ are derived according to (4.42). For a fixed element on the global scale, the virtual displacements, the tangential element stiffness matrix and the element residuum vector are denoted by $\delta\mathbf{v}^G, \mathbf{k}^G$ and $\mathbf{f}^G$, respectively. The global-local coupling is given by $\mathbf{k}^G = \mathbf{k}^G(\mathbf{D}_i)$ and $\mathbf{f}^G = \mathbf{f}^G(\boldsymbol{\sigma}_i)$, which can be written in the following form:

$$\begin{aligned}\mathbf{k}^G(\mathbf{D}_i) &= \int\limits_{(\Omega_e)} (\mathbf{B}^T\mathbf{D}\mathbf{B} + \mathbf{G})\,\mathrm{d}A \\ \mathbf{f}^G(\boldsymbol{\sigma}_i) &= \int\limits_{(\Omega_e)} (\mathbf{B}^T\boldsymbol{\sigma} - \mathbf{N}^T\bar{\mathbf{p}})\,\mathrm{d}A - \int\limits_{(\Gamma_{e\sigma})} \mathbf{N}^T\bar{\mathbf{t}}\,\mathrm{d}s\end{aligned} \tag{5.7}$$

Here, $\mathbf{N}$, $\mathbf{B}$ and $\mathbf{G}$ are used according to (4.2), (4.36), and (4.38), respectively, while the vectors of stress resultants $\boldsymbol{\sigma}_i$ and matrices of linearized stress resultants $\mathbf{D}_i$ will be specified in the following.

Using (5.6) and (5.7), the coupled linearized weak form (5.5) can be expressed in terms of displacements, virtual displacements and linearized displacements, with e denoting global-scale element numbers and i describing a specific integration point in such an element:

$$L\left[g(\boldsymbol{\theta}^h, \delta\boldsymbol{\theta}^h), \Delta\boldsymbol{\theta}^h\right] =$$

$$\sum_{e=1}^{numel} \begin{bmatrix} \delta\mathbf{v}^G \\ \delta\mathbf{V}_1 \\ \vdots \\ \delta\mathbf{V}_i \\ \vdots \\ \delta\mathbf{V}_{ngp} \end{bmatrix}_e^T \left\{ \begin{bmatrix} \mathbf{k}^G(\mathbf{D}_i) & & & & \dots & 0 \\ & \mathbf{K}_1^L & & & & \vdots \\ & & \ddots & & & \\ & & & \mathbf{K}_i^L & & \\ \vdots & & & & \ddots & \\ 0 & \dots & & & & \mathbf{K}_{ngp}^L \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v}^G \\ \Delta\mathbf{V}_1 \\ \vdots \\ \Delta\mathbf{V}_i \\ \vdots \\ \Delta\mathbf{V}_{ngp} \end{bmatrix} + \begin{bmatrix} \mathbf{f}^G(\boldsymbol{\sigma}_i) \\ \mathbf{F}_1^L \\ \vdots \\ \mathbf{F}_i^L \\ \vdots \\ \mathbf{F}_{ngp}^L \end{bmatrix} \right\}_e \tag{5.8}$$

The local element displacement vector $\mathbf{v}_e$ in (5.6) is now partitioned into two parts. $\mathbf{v}_\Omega =: \mathbf{v}_a$ contains displacements at inner nodes, whereas $\mathbf{v}_\Gamma =: \mathbf{v}_b$ comprises the displacements at the boundary of the RVE. Using a standard assembly matrix $\mathbf{a}_e$, the internal displacements $\mathbf{v}_a$ are related to the global displacement vector $\mathbf{V}_i$. On the other hand, the boundary displacements $\mathbf{v}_b$ follow from the prescribed shell strains $\boldsymbol{\varepsilon}_i$ of the global shell model in the integration point i. The relation is given by a matrix $\mathbf{A}_e(x, y, z)$ which will be introduced later.

$$\mathbf{v}_e = \begin{bmatrix} \mathbf{v}_\Omega \\ \mathbf{v}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_b \end{bmatrix} = \begin{bmatrix} \mathbf{a}_e\mathbf{V}_i \\ \mathbf{A}_e\boldsymbol{\varepsilon}_i \end{bmatrix} \tag{5.9}$$

The associated variations $\delta\mathbf{v}_e$ and linearizations $\Delta\mathbf{v}_e$ immediately follow:

$$\delta\mathbf{v}_e = \begin{bmatrix} \delta\mathbf{v}_a \\ \\ \delta\mathbf{v}_b \end{bmatrix} = \begin{bmatrix} \mathbf{a}_e \delta\mathbf{V}_i \\ \\ \mathbf{A}_e \delta\boldsymbol{\varepsilon}_i \end{bmatrix}, \quad \Delta\mathbf{v}_e = \begin{bmatrix} \Delta\mathbf{v}_a \\ \\ \Delta\mathbf{v}_b \end{bmatrix} = \begin{bmatrix} \mathbf{a}_e \Delta\mathbf{V}_i \\ \\ \mathbf{A}_e \Delta\boldsymbol{\varepsilon}_i \end{bmatrix} \tag{5.10}$$

According to the partitioning (5.9), the element tangential stiffness matrix $\mathbf{k}_e^L$ and residuum vector $\mathbf{f}_e^L$ are partitioned in the same way,

$$\mathbf{k}_e^L = \begin{bmatrix} \mathbf{k}_{aa} & \mathbf{k}_{ab} \\ \\ \mathbf{k}_{ba} & \mathbf{k}_{bb} \end{bmatrix}_e, \quad \mathbf{f}_e^L = \begin{bmatrix} \mathbf{f}_a \\ \\ \mathbf{f}_b \end{bmatrix}_e \quad . \tag{5.11}$$

Inserting these relations into (5.6) yields

$$\begin{aligned} &\delta\mathbf{V}_i^T(\mathbf{K}_i^L \Delta\mathbf{V}_i + \hat{\mathbf{F}}_i^L) \\ &\quad = \frac{1}{A_i} \sum_{e=1}^{n} \begin{bmatrix} \delta\mathbf{v}_a \\ \\ \delta\mathbf{v}_b \end{bmatrix}_e^T \left\{ \begin{bmatrix} \mathbf{k}_{aa} & \mathbf{k}_{ab} \\ \\ \mathbf{k}_{ba} & \mathbf{k}_{bb} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v}_a \\ \\ \Delta\mathbf{v}_b \end{bmatrix} + \begin{bmatrix} \mathbf{f}_a \\ \\ \mathbf{f}_b \end{bmatrix} \right\}_e \\ &\quad = \frac{1}{A_i} \sum_{e=1}^{n} \begin{bmatrix} \delta\mathbf{V}_i \\ \\ \delta\boldsymbol{\varepsilon}_i \end{bmatrix}_e^T \left\{ \begin{bmatrix} \mathbf{a}_e^T \mathbf{k}_{aa} \mathbf{a}_e & \mathbf{a}_e^T \mathbf{k}_{ab} \mathbf{A}_e \\ \\ \mathbf{A}_e^T \mathbf{k}_{ba} \mathbf{a}_e & \mathbf{A}_e^T \mathbf{k}_{bb} \mathbf{A}_e \end{bmatrix} \begin{bmatrix} \Delta\mathbf{V}_i \\ \\ \Delta\boldsymbol{\varepsilon}_i \end{bmatrix} + \begin{bmatrix} \mathbf{a}_e^T \mathbf{f}_a \\ \\ \mathbf{A}_e^T \mathbf{f}_b \end{bmatrix} \right\}_e \quad . \end{aligned} \tag{5.12}$$

By introducing the quantities

$$\begin{aligned} \mathbf{K} &= \sum_{e=1}^{n} \mathbf{a}_e^T \mathbf{k}_{aa} \mathbf{a}_e & \hat{\mathbf{F}}_a &= \sum_{e=1}^{n} \mathbf{a}_e^T \mathbf{f}_a \\ \mathbf{L} &= \sum_{e=1}^{n} \mathbf{a}_e^T \mathbf{k}_{ab} \mathbf{A}_e & \hat{\mathbf{F}}_b &= \sum_{e=1}^{n} \mathbf{A}_e^T \mathbf{f}_b \\ \mathbf{M} &= \sum_{e=1}^{n} \mathbf{A}_e^T \mathbf{k}_{bb} \mathbf{A}_e \quad , \end{aligned} \tag{5.13}$$

equation (5.12) can be written as

$$\delta\mathbf{V}_i^T(\mathbf{K}_i^L \Delta\mathbf{V}_i + \hat{\mathbf{F}}_i^L) = \frac{1}{A_i} \begin{bmatrix} \delta\mathbf{V}_i \\ \\ \delta\boldsymbol{\varepsilon}_i \end{bmatrix}^T \left\{ \begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \\ \mathbf{L}^T & \mathbf{M} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{V}_i \\ \\ \Delta\boldsymbol{\varepsilon}_i \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{F}}_a \\ \\ \hat{\mathbf{F}}_b \end{bmatrix} \right\} \quad . \tag{5.14}$$

As each row of the matrix in (5.8) must vanish, one obtains with an arbitrary variation $\delta\mathbf{V}_i \neq 0$ the equation

$$\mathbf{K}\Delta\mathbf{V}_i + \mathbf{L}\Delta\boldsymbol{\varepsilon}_i + \hat{\mathbf{F}}_a = 0 \tag{5.15}$$

which allows to eliminate the internal degrees of freedom $\Delta\mathbf{V}_i$ by expressing them in terms of the other parameters:

$$\Delta\mathbf{V}_i = -\mathbf{K}^{-1}(\mathbf{L}\Delta\boldsymbol{\varepsilon}_i + \hat{\mathbf{F}}_a) \tag{5.16}$$

Regularity of $\mathbf{K}$ is given as potential rigid body motions are restricted by appropriate boundary conditions. The previously mentioned stress resultants $\boldsymbol{\sigma}_i$ and linearized stress resultants $\mathbf{D}_i$ are now introduced as

$$\boldsymbol{\sigma}_i = \frac{1}{A_i}(\hat{\mathbf{F}}_b - \mathbf{L}^T\mathbf{K}^{-1}\hat{\mathbf{F}}_a), \qquad \mathbf{D}_i = \frac{1}{A_i}(\mathbf{M} - \mathbf{L}^T\mathbf{K}^{-1}\mathbf{L}) \quad . \tag{5.17}$$

Together with (5.16), equation (5.14) reduces to

$$\begin{aligned}\delta\mathbf{V}_i^T(\mathbf{K}_i^L\Delta\mathbf{V}_i + \hat{\mathbf{F}}_i^L) &= \frac{1}{A_i}\delta\boldsymbol{\varepsilon}_i^T\left[(\mathbf{M} - \mathbf{L}^T\mathbf{K}^{-1}\mathbf{L})\Delta\boldsymbol{\varepsilon}_i + (\hat{\mathbf{F}}_b - \mathbf{L}^T\mathbf{K}^{-1}\hat{\mathbf{F}}_a)\right] \\ &= \delta\boldsymbol{\varepsilon}_i^T(\mathbf{D}_i\Delta\boldsymbol{\varepsilon}_i + \boldsymbol{\sigma}_i) \quad .\end{aligned} \tag{5.18}$$

Inserting this representation back into (5.8) leads to the following formulation:

$$L\left[g(\boldsymbol{\theta}^h, \delta\boldsymbol{\theta}^h), \Delta\boldsymbol{\theta}^h\right] =$$

$$\sum_{e=1}^{numel} \begin{bmatrix} \delta\mathbf{v}^G \\ \delta\boldsymbol{\varepsilon}_1 \\ \vdots \\ \delta\boldsymbol{\varepsilon}_i \\ \vdots \\ \delta\boldsymbol{\varepsilon}_{ngp} \end{bmatrix}_e^T \left\{ \begin{bmatrix} \mathbf{k}^G(\mathbf{D}_i) & & & \dots & 0 \\ & \mathbf{D}_1 & & & \vdots \\ & & \ddots & & \\ & & & \mathbf{D}_i & \\ \vdots & & & \ddots & \\ 0 & \dots & & & \mathbf{D}_{ngp} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v}^G \\ \Delta\boldsymbol{\varepsilon}_1 \\ \vdots \\ \Delta\boldsymbol{\varepsilon}_i \\ \vdots \\ \Delta\boldsymbol{\varepsilon}_{ngp} \end{bmatrix} + \begin{bmatrix} \mathbf{f}^G(\boldsymbol{\sigma}_i) \\ \boldsymbol{\sigma}_1 \\ \vdots \\ \boldsymbol{\sigma}_i \\ \vdots \\ \boldsymbol{\sigma}_{ngp} \end{bmatrix} \right\}_e \tag{5.19}$$

In this notation, the bidirectional global-local coupling becomes evident, as the local shell strains of the global shell model $\boldsymbol{\varepsilon}_i$ enter into the local systems $i = 1, \dots, ngp$ and on the other hand, the ensuing stress resultants $\boldsymbol{\sigma}_i$ and linearized stress resultants $\mathbf{D}_i$ are processed in the global system, corresponding to the first row of the matrix. It is noted that all ngp local systems in a single global element are independent from each other and can thus be solved in parallel. In the same way, all local systems belonging to other global elements e are independent, allowing for massive parallelization. The coupled nonlinear system is solved with an ordinary Newton iteration scheme, where equilibrium has to be attained both locally in all integration points, as well as globally for the shell.

5.3 Boundary conditions for the RVE

The boundary conditions for the representative volume element must be specified in a way to fulfill the Hill condition [27, 32], stating that the microscopic and macroscopic stress power must be equal. In this work, displacement boundary conditions are used at the lateral boundaries of the RVE, whereas stress boundary conditions are present at the upper and lower surfaces. An arbitrary RVE is discretized in space with Euclidean coordinates

$$x \in [-l_x/2, l_x/2], \quad y \in [-l_y/2, l_y/2], \quad z \in [-h/2, h/2] \quad . \tag{5.20}$$

For an illustrative example of an RVE and its boundary conditions, see Figures 6.1, 6.2 in the next chapter. The prescribed boundary displacements $\bar{\mathbf{u}}$ can be specified in terms of the Green-Lagrange strains $\bar{\mathbf{E}}$ of the global model, assuming small strains:

$$\begin{bmatrix} \bar{u}_x \\ \bar{u}_y \\ \bar{u}_z \end{bmatrix} = \begin{bmatrix} \bar{E}_{11} & \bar{E}_{12} & \bar{E}_{13} \\ \bar{E}_{21} & \bar{E}_{22} & \bar{E}_{23} \\ \bar{E}_{31} & \bar{E}_{32} & \bar{E}_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.21}$$

The averaged strains $\bar{\mathbf{E}}$ can in turn be related to the shell strains $\boldsymbol{\varepsilon}$ using (3.20) with $z = \xi^3$, which yields

$$\begin{bmatrix} \bar{u}_x \\ \bar{u}_y \\ \bar{u}_z \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} + z\kappa_{11} & \varepsilon_{12} + z\kappa_{12} & \gamma_1/2 \\ \varepsilon_{12} + z\kappa_{12} & \varepsilon_{22} + z\kappa_{22} & \gamma_2/2 \\ \gamma_1/2 & \gamma_2/2 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad . \tag{5.22}$$

It is observed in [27] that the prescribed displacement $\bar{u}_z$ represents severe constraints for particular deformation modes of the RVE, especially for torsion modes $2\kappa_{12}$. As the RVEs used in this work behave in the same way, the boundary conditions are modified accordingly, freeing the displacements $\bar{u}_z$:

$$\begin{bmatrix} \bar{u}_x \\ \bar{u}_y \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} + z\kappa_{11} & \varepsilon_{12} + z\kappa_{12} & \gamma_1 \\ \varepsilon_{12} + z\kappa_{12} & \varepsilon_{22} + z\kappa_{22} & \gamma_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.23}$$

This equation can be rewritten in terms of the vector of shell strains $\boldsymbol{\varepsilon}$. One obtains

$$\begin{bmatrix} \bar{u}_x \\ \bar{u}_y \end{bmatrix} = \begin{bmatrix} x & 0 & \frac{1}{2}y & xz & 0 & \frac{1}{2}yz & z & 0 \\ 0 & y & \frac{1}{2}x & 0 & yz & \frac{1}{2}xz & 0 & z \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \\ \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \\ \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{5.24}$$

or in a shorter notation

$$\bar{\mathbf{u}}_I = \mathbf{A}_I(x, y, z)\, \boldsymbol{\varepsilon} \quad . \tag{5.25}$$

The collection of matrices $\mathbf{A}_I$ then describes the matrix $\mathbf{A}_e$ introduced in (5.9)

$$\mathbf{A}_e = \begin{bmatrix} \delta_1 \mathbf{A}_1 \\ \vdots \\ \delta_I \mathbf{A}_I \\ \vdots \\ \delta_{nel} \mathbf{A}_{nel} \end{bmatrix} \quad \text{with} \quad \delta_I = \begin{cases} 1 & \text{if node } I \text{ has fixed DOFs} \\ 0 & \text{else} \end{cases} \quad , \tag{5.26}$$

where $nel = 4$ describes the number of nodes per element in the local discretization. Additionally, due to not prescribing displacements $\bar{u}_z$, link conditions have to be set for the out-of-plane displacements to prevent rigid body rotations when applying shear strains γ_α. The nodal displacements on the lateral boundaries $x = \pm l_x/2$ and $y = \pm l_y/2$ are linked in an antisymmetric way via

$$\begin{aligned} \bar{u}_z(l_x/2, y, z) &= \bar{u}_z(-l_x/2, -y, z) \\ \bar{u}_z(x, l_y/2, z) &= \bar{u}_z(-x, -l_y/2, z) \quad . \end{aligned} \tag{5.27}$$

As a final measure, an arbitrary node must be fixed in z direction to prevent rigid body motions along the z axis. If available, a center node at $(x, y, z) = 0$ is chosen.

5.4 Simultaneous iteration

The coupled nonlinear system is solved in a Newton-type iteration with incremental load steps as described in Chapter 4. It has been shown in (5.19) that all local boundary value problems corresponding to integration points in the global model are independent of each other, allowing to find approximate solutions in parallel for all local equation systems. In the case of nonlinearity in the local systems, a full Newton iteration has to be conducted, usually requiring several iteration steps before attaining local equilibrium. Carrying out multiple local iteration steps in each global iteration of a single load step evidently leads to high calculation times.

It was shown in [27], however, that local equilibrium can also be attained when only performing a single local iteration step for each global iteration step, without requiring additional iterations for the global system. The quadratic convergence rate is maintained. This so-called simultaneous iteration was utilized for the examples in this work as well.

Chapter 6

Numerical modeling of sandwich structures

In this work, different types of sandwich cores will be investigated and approximate solutions for global structures using these core layouts on a local scale will be numerically obtained in terms of the coupled model, leading to several different configurations for the representative volume element. As a first step, it must be shown that the developed homogenization method yields valid results for the linearized stress resultants $\mathbf{D}_i$ in an RVE, see (5.17). The material tangent matrix must be independent of the exact choice of an RVE, as long as it reflects the microstructure of the current system and the finite element mesh is sufficiently fine. In some cases, it is possible to verify the numerical results for simple RVE configurations using analytical expressions. For more complicated microstructures, the material matrices stemming from different admissible RVEs are compared to each other.

All investigated RVEs consist of upper and lower face layers, representing surfaces parallel to the global shell reference surface at $z = \pm h/2$, where h denotes both the global shell thickness as well as the height of the RVE. Interior stiffeners are placed perpendicularly to the face layers, either along one or both of the in-plane coordinate directions. In addition, hexagonal honeycomb structures are modeled with cell wall segments perpendicular to the reference surface in three different directions.

Once it has been shown for each type of microstructure that the RVEs lead to converging linearized stress resultants, coupled simulations will be carried out where the given RVEs are used to describe the microstructure of a larger global system. The simulation results of all coupled models are compared to finely discretized full scale shell models, where the complete microstructure of a sandwich is modeled using classical shell elements with intersections. These reference solutions lead to very large equation systems with up to several million unknowns even for comparably simple academic examples.

Most calculations, especially the coupled simulations and the investigation of different

RVEs, will be carried out using FEAP. For some of the full scale reference solutions, however, the capacities of FEAP are exceeded in terms of the equation system size. In these cases, the commercial FEM tool Abaqus [1] will be employed to generate the reference solutions with a classical shell model.

6.1 Parameter terminology

Since many numerical examples will be investigated and compared in this chapter, an unambiguous terminology for geometry, discretization and material parameters has to be established. Most of these parameters can be generalized for all considered models and are summarized in Table 6.1. Throughout the whole chapter, $[kN]$ and $[cm]$ will be presumed as consistent base units for all force and length parameters, if not explicitly stated otherwise. Consequently, derived units will be used without further introduction as well, e.g. $[kN/cm^2]$ for stresses.

Some parameters only apply to certain examples, such as the material parameters E, ν, and Y_0, of which E and ν are used for all materials but Y_0 only applies to the examples with elasto-plastic material behavior. Likewise, the external load q_z represents either a line load (for the unidirectionally stiffened structure) or an area load (for the bidirectionally stiffened and honeycomb structures). It is noted that all loads q_z are applied to the top surface of the structure, corresponding to the upper face layer in the full scale models. This is not possible for the coupled model, where loads q_z are applied as nodal forces in the shell element nodes of the global system.

6.2 Unidirectionally stiffened sandwich panel

As a first motivational example a sandwich structure is considered which consists of two face layers and stiffeners (also called cell walls in a broader sense) parallel to one axis direction, perpendicular to the face layers. Without loss of generality, it is assumed that the stiffeners are placed alongside or parallel to the x-axis where $y = \text{const}$, so that $n_y \geq 1$ is given.

6.2.1 Validation of RVE material matrix

The initial step for each type of cell structure is the validation of the results for the RVE material matrix of a chosen exemplary microstructure configuration. For this example, an

param.	description	unit
l_x	Length of the full structure in x-direction.	$[cm]$
l_y	Length of the full structure in y-direction.	$[cm]$
h	Height of the full structure (length in z-direction). Equal to the RVE height.	$[cm]$
t	Thickness of shells. If not explicitly stated otherwise, refers to the thickness of both face layers and the cell walls.	$[cm]$
t_{fl}	Thickness of face layers, if not equal to the thickness of cell walls.	$[cm]$
t_{cw}	Thickness of cell walls, if not equal to the thickness of face layers.	$[cm]$
n_x	Number of cell walls with $x = \text{const}$ (parallel to the y-axis) in the full structure.	$[-]$
n_y	Number of cell walls with $y = \text{const}$ (parallel to the x-axis) in the full structure.	$[-]$
h_x	Discretization fineness, number of elements between two cell walls with $x = \text{const}$.	$[-]$
h_y	Discretization fineness, number of elements between two cell walls with $y = \text{const}$.	$[-]$
h_z	Discretization fineness, number of elements in thickness direction between the two face layers.	$[-]$
l_x^R	Length of RVE in x-direction.	$[cm]$
l_y^R	Length of RVE in y-direction.	$[cm]$
n_x^R	Number of cell wall segments with $x = \text{const}$ (parallel to the y-axis) in an RVE.	$[-]$
n_y^R	Number of cell wall segments with $y = \text{const}$ (parallel to the x-axis) in an RVE.	$[-]$
h_x^R	RVE discretization fineness, number of elements between two cell walls with $x = \text{const}$.	$[-]$
h_y^R	RVE discretization fineness, number of elements between two cell walls with $y = \text{const}$.	$[-]$
h_z^R	RVE discretization fineness, number of elements in thickness direction between the two face layers.	$[-]$
E	Young's modulus (all materials)	$[kN/cm^2]$
ν	Poisson ratio (all materials)	$[-]$
Y_0	Initial yield stress (elasto-plastic material only)	$[kN/cm^2]$
q_z	Constant line load (unidirectionally stiffened structure) or area load (bidirectionally stiffened and honeycomb structures)	$[kN/cm]$ or $[kN/cm^2]$

Table 6.1: Summary of model parameters for the numerical examples in this chapter.

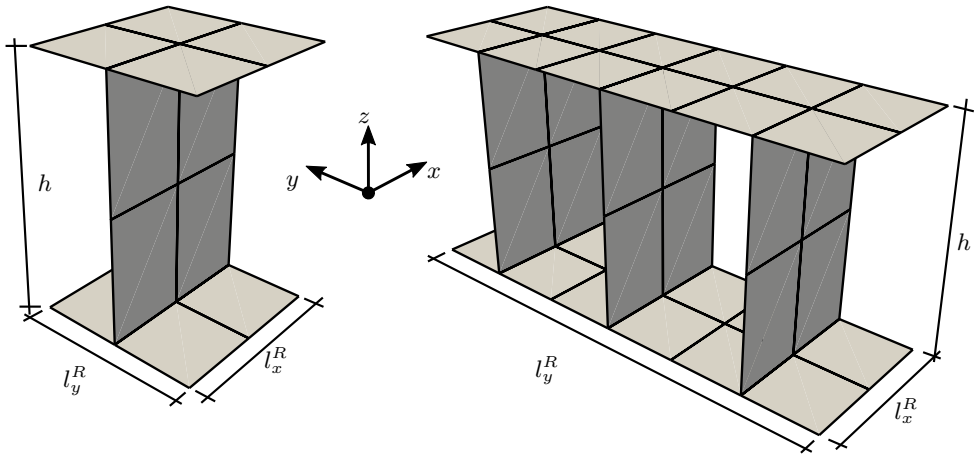


Figure 6.1: Representative volume elements with $n_y^R = 1$ (left) and $n_y^R = 3$ (right) stiffeners.

arrangement of parallel cell walls placed in intervals of $2\,[cm]$ is assumed. Specifically, an RVE with measurements $l_y^R = 2 \cdot n_y^R\,[cm]$, $h = 3\,[cm]$ and local shell thickness $t = 0.1\,[cm]$ is considered. The origin of this and all following RVEs is chosen to be in the center of the structure, so that the RVE domain is bounded by the cuboid

$$[-l_x^R/2, l_x^R/2] \times [-l_y^R/2, l_y^R/2] \times [-h/2, h/2] \subset \mathbb{R}^3 \quad . \tag{6.1}$$

Two possible discretizations are shown in Figure 6.1, whereas the in-plane boundary conditions (5.24) and link conditions (5.27) of the RVE are depicted in Figure 6.2. Additionally, the local drilling degree of freedom is fixed on all nodes except for the intersection nodes where a stiffener is connected to the upper or lower face layer. These nodes reside on lines $(x, y_I, \pm h/2)$, where $x \in [-l_x^R/2, l_x^R/2]$ and y_I, $(I = 1, \ldots, n_y^R)$, denotes the constant y-coordinate for one of the n_y^R stiffeners.

As a first step, the matrix of linearized stress resultants is investigated for various choices of suitable RVEs. For a homogeneous shell, the Reissner-Mindlin theory requires decoupled stiffness terms for membrane, bending and shear modes, so that the material matrix, containing the linearized stress resultants, would assume the structure

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^m & \mathbf{D}^{mb} & 0 \\ \mathbf{D}^{mb} & \mathbf{D}^b & 0 \\ 0 & 0 & \mathbf{D}^s \end{bmatrix}_{(8\times 8)} \quad . \tag{6.2}$$

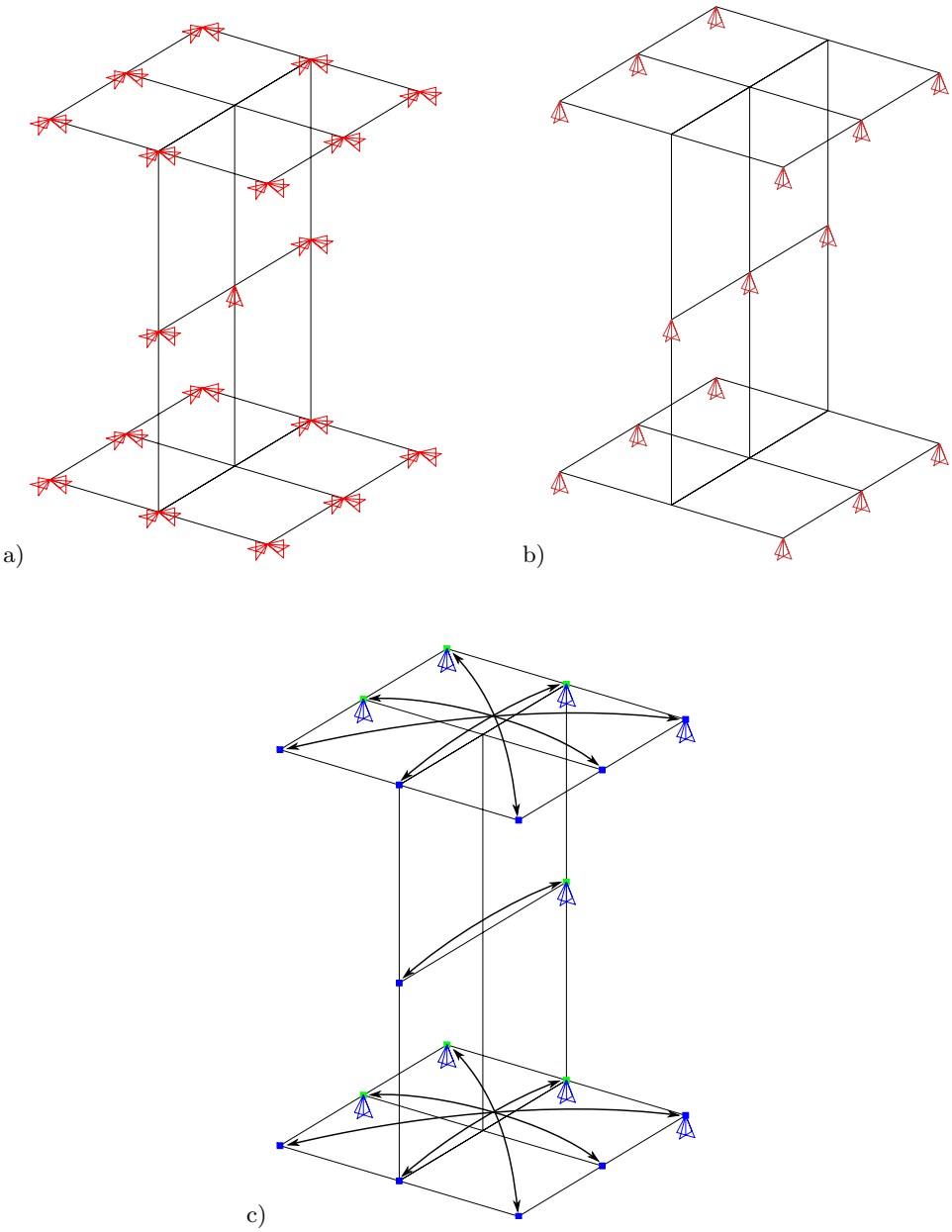


Figure 6.2: (a) In-plane locked boundary conditions for u_x, u_y and locked boundary condition for u_z on a single node to prevent rigid body motions.
(b) Fixed 6th degree of freedom except on intersection nodes.
(c) Antisymmetric out-of-plane displacement link conditions for u_z, for which master nodes are drawn as a support.

It will be seen that this special structure can be preserved in most of the examples for the linear case, using uni- or bidirectionally stiffened core structures. However, when more complicated hexagonal structures or physical nonlinearity are introduced later, the decoupled structure of $\mathbf{D}$ is lost. Furthermore, it is important to investigate the behavior of $\mathbf{D}$ for different possible RVE configurations for a given microstructure. In theory, all admissible RVEs in terms of micromechanics, for given shell strains $\boldsymbol{\varepsilon}$, are required to lead to the same material matrix $\mathbf{D}$, regardless of any free parameters, which are in this case l_x^R, n_y^R and the chosen finite element discretization. While it is evidently impossible to exactly fulfill this assumption in an applied numerical simulation, at least a good approximation must be established. For the unidirectionally stiffened shell the diagonal components $D_{11}, \dots D_{55}$ of $\mathbf{D}$ can be evaluated analytically for linear-elastic material behavior with Young's modulus E and Poisson ratio $\nu = 0$. In this case, it holds for $t \ll l_x^R, l_y^R$

$$\begin{aligned} D_{11} &= \frac{EA_y}{l_y^R} = \frac{E(2 \cdot l_y^R t + n_y^R \cdot ht)}{l_y^R} \\ D_{22} &= \frac{EA_x}{l_x^R} = 2 \cdot Et \\ D_{44} &= \frac{EI_y}{l_y^R} = \frac{E}{l_y^R} \cdot \left(\frac{h^2}{2} l_y^R t + \frac{n_y^R}{12} h^3 t \right) \\ D_{55} &= \frac{EI_x}{l_x^R} = \frac{1}{2} E h^2 t \quad , \end{aligned} \tag{6.3}$$

where A_x and A_y are cross section areas in planes parallel to the x- and y-axes, respectively, and I_y, I_z are the area moments of inertia, calculated as for a common I-beam. It is noted that A_y and I_y contain terms for both the face layers as well as the stiffener(s), whereas only the face layers contribute to A_x and I_x. For a quadratic reference surface with $l^R := l_x^R = l_y^R$, it is possible to additionally obtain

$$D_{33} = \frac{GA}{l^R} = 2 \cdot Gt \tag{6.4}$$

with $A := A_x = 2 \cdot l^R t$. For D_{66}, D_{77} and D_{88}, no trivial analytical solution was found.

A numerical simulation is carried out for $l_x^R = 2$, $n_y^R = 1$ and thus $l_y^R = 2$, with material properties $E = 7000\,[kN/cm^2]$, $\nu = 0$, $G = E/2$. A discretization of 2 elements per side is chosen, as illustrated in Figure 6.1(a). The results show that the material matrix $\mathbf{D}$ is in fact completely decoupled and has diagonal structure, $\mathbf{D} = \mathrm{diag}(D_{11}, \dots, D_{88})$.

The analytical and numerical results are compared in Table 6.2, where a very good approximation for D_{44} and exact values for D_{11}, D_{22}, D_{33} and D_{55} can be seen. It is observed that the membrane and bending components $D_{11}, \dots, D_{66}$ require no mesh

component	D_{11}	D_{22}	D_{33}	D_{44}	D_{55}	D_{66}	D_{77}	D_{88}
analytical	2450	1400	700	3937.5	3150	-	-	-
FE $2 \times 2 \times 2$	2450	1400	700	3938.32	3150	1575.52	495.97	1.16
FE $64 \times 64 \times 64$	2450	1400	700	3938.24	3150	1575.57	479.88	1.93

Table 6.2: Components of the material matrix $D_{11}, \ldots, D_{88}$ for the unidirectionally stiffened RVE with $l_x^R = 2$ and different discretizations h_x^R, h_y^R, h_z^R.

refinement, while shear stiffnesses D_{77} and D_{88} show discrepancies of 3.35% and 39.63%, respectively, when comparing the minimal discretization with 2 elements per side to a fully converged solution. In (6.3) and (6.4), it is evident that the analytical solutions for stiffness components $D_{11}, \ldots, D_{55}$ are independent of l_x^R and thus allow arbitrary RVE lengths to be chosen. Indeed, calculations with varying l_x^R yield the same values for these components. For D_{66} and D_{88}, nearly constant values are obtained when varying $l_x^R = 1, \ldots, 100\,[cm]$ with relative differences of at most 1.2%. However, for D_{77} different values are obtained for varying lengths as a result of the RVE boundary link conditions.

By introducing a modified shear correction factor κ, depending on the length-to-height ratio $l_x^R \cdot l_y^R / h^2$ of the RVE, this effect can be mitigated, leading to components D_{77} which only differ up to roughly 20% from each other for very short or very long RVEs. In comparison, the link conditions are still superior to alternative boundary conditions for the out-of-plane displacements, which according to (5.22) lead to wrong stiffness parameters in certain membrane and bending modes, as was shown in [27]. Particularly, torsion strains $2\kappa_{12}$ would be severely restrained, which is not the case when using link conditions. The results for the torsion and shear stiffness components can be seen in Figure 6.3(a) for varying RVE lengths l_x^R, whereas Figure 6.3(b) shows the unrestrained deformation resulting from a torsion strain.

While it was shown that the proposed RVE for the unidirectionally stiffened sandwich shell is somewhat sensible to length scaling and discretization in its shear stiffnesses, the membrane and bending modes fulfill all requirements and the matrix of linearized stress resultants possesses a completely decoupled structure. It will be seen in the next subsection that, despite the only roughly accurate shear terms, good results can be obtained for coupled global-local systems using the present RVE even for moderately thin structures.

For materials with $\nu > 0$, the diagonal structure of $\mathbf{D}$ is lost, but membrane, bending and shear modes are still decoupled as only D_{12} and D_{45} appear as new terms on the secondary diagonal.

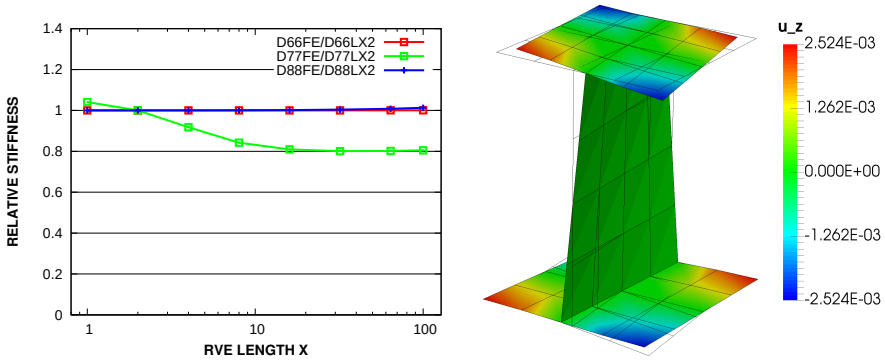


Figure 6.3: (a) Torsional and shear stiffness components for different RVE lengths l_x^R, compared to results for $l_x^R = 2$.
(b) Mesh and deformed mesh (10x amplified) for torsion strain $2\kappa_{12} = 0.02$.

6.2.2 Axially stiffened sandwich beam

As a first coupled example for the unidirectionally stiffened microstructure, a beam-like sandwich structure is considered, consisting of two face layers ($l_x = 100, l_y = 6, t = 0.1$) at $z = \pm h/2$ with $h = 3$, as well as $n_y = 12$ axial stiffeners with the same length l_x, height h, and thickness t. Contrary to the last section, the distance between two neighboring cell walls follows as $l_y/n_y = 0.5\,[cm]$. The origin is chosen to be in the center of the structure, so that its domain is bounded by the cuboid

$$[-l_x/2, l_x/2] \times [-l_y/2, l_y/2] \times [-h/2, h/2] \subset \mathbb{R}^3 \quad . \tag{6.5}$$

Linear-elastic material behavior is assumed with constants $E = 7000$, $\nu = 0$. Additionally, linear geometry is presumed. In Figure 6.4, an exemplary discretization using generic 6-DOF shell elements with intersections is illustrated for a system with $n_y = 4$ stiffeners. It is noted that the model depicted in this figure does not represent the aforementioned choice of parameters, in order to provide a clearer illustration. The corresponding full scale systems will also serve as reference solutions to validate later results from the coupled model.

The structure is supported at the end nodes fulfilling $x = \pm l_x/2, z = 0$ with Navier-type boundary conditions, where only the 3rd translational degree of freedom u_z is fixed. Additionally, at two single nodes u_x and at one node u_y are fixed to prevent rigid body motions, while the 6th degree of freedom is locked on all non-intersection nodes. Two constant line loads $q_z = 0.1$ are applied at $x = \pm l_x/6$ on the upper surface $z = h/2$. The

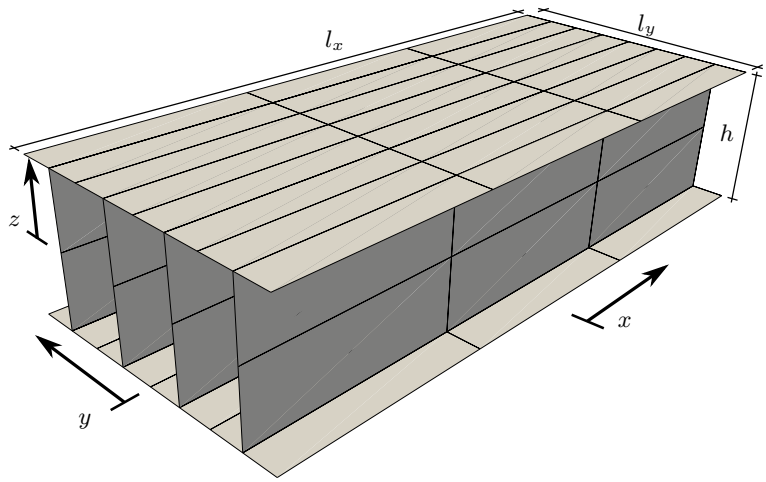


Figure 6.4: Exemplary discretization of an axially stiffened beam with $n_y = 4$ stiffeners.

full system with boundary conditions and external loads is illustrated in Figure 6.5(a).

For this comparably simple system, an analytic solution for the beam's displacements can be found by means of the classical beam theory. Both face layers and the stiffeners contribute to the structure's area moment of inertia. For small t,

$$I_y = \frac{h^2}{2} l_y t + \frac{n_y}{12} h^3 t = 5.4 \tag{6.6}$$

is given using the above parameters. The beam's maximum vertical displacement is obtained for the given load and support configuration:

$$w_{\max} = w_b + w_s = \frac{23}{648} \frac{q_z l_x^3 l_y}{E I_y} + \frac{q_z l_x l_y}{3\,G A_s} = 0.5634 + 0.0016 = 0.5650 \tag{6.7}$$

with $A_s \approx n_y \cdot ht = 3.6$ corresponding to the cross section area of the stiffeners. In order to derive the coupled formulation of the problem, a representative volume element is introduced. It is expedient to choose a small and coarsely discretized RVE to minimize calculation time, while still preserving correctness of the results. Thus, an RVE with a single stiffener $n_y^R = 1$, where $(.)^R$ indicates quantities relating to the RVE, and 2 elements per side is chosen, as shown before in Figure 6.2 together with applicable boundary and link conditions. The RVE height $h^R = h = 3$ corresponds to the height of the global structure, whereas the width l_y^R is given so that

$$\frac{l_y^R}{l_y} = \frac{n_y^R}{n_y} \tag{6.8}$$

holds, yielding $l_y^R = 0.5$ in this case. The length l_x^R is arbitrarily chosen as $l_x^R = l_y^R = 0.5$. The global part of the coupled system is discretized using shell elements with the full thickness h. As no intersections are present on the global scale, shell elements with 5 degrees of freedom can be employed. In the integration points of the global shell elements, the vector of shell strains is evaluated and leads to prescribed deformations for the RVE nodes according to (5.24). Figure 6.5 illustrates the classical full scale model of the beam structure and the global structure of the coupled global-local model in comparison. It can be seen that the coupled model naturally comprises significantly less elements.

In the depicted discretizations with 24 elements in x-direction, leading to converged results for both models, the full model contains 1728 elements and 8323 equations, whereas the coupled model on the global scale only comprises 24 elements and 243 equations. Adding to that 24×4 required RVE evaluations with 12 elements and 65 equations each, the coupled system is potentially much faster to solve with 1×243 and 96×65 independent equations, due to the quadratically increasing numeric effort for directly solving sparse linear systems.

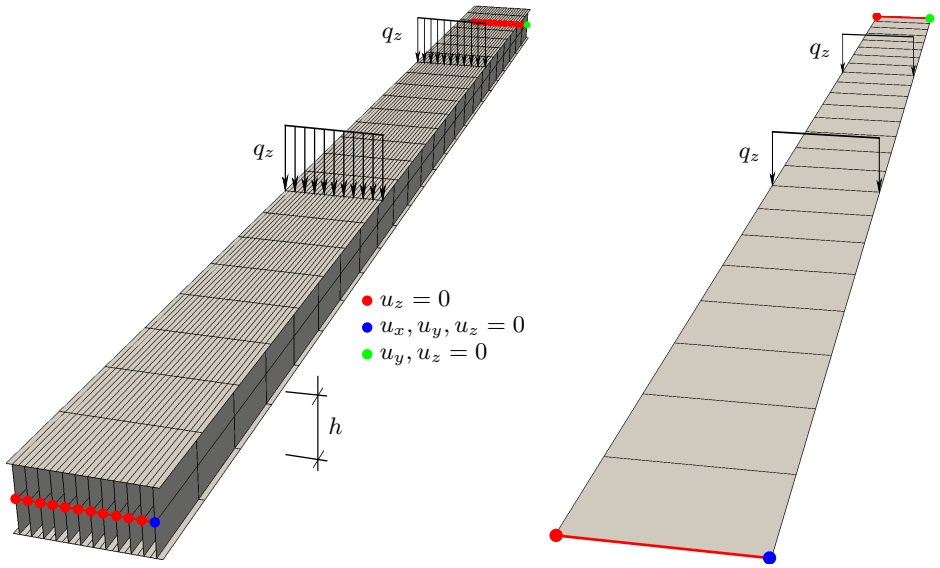
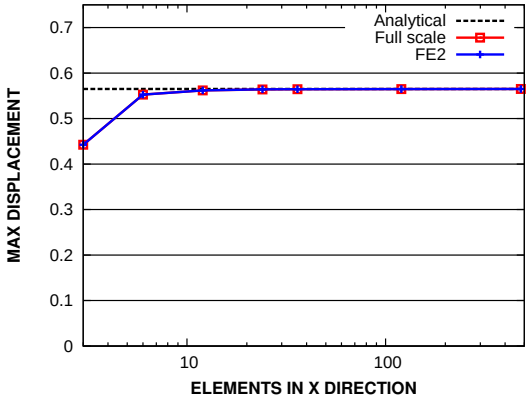


Figure 6.5: (a) Full scale model of the beam structure, using classical 6-DOF shell elements with intersections.
(b) Global scale model of the coupled system, using non-intersected shell elements with thickness h.

Both finite element models lead to a fully converged solution of $w_{\max} = 0.5650$ when using enough elements along the x-direction, see Figure 6.6(a). It is noted that additional elements along the y- or z-directions or in the discretization of the RVE have no influence

on the solution. Additionally, the chosen minimal RVE with $n_y = 1$ is sufficient as larger RVEs with a higher number of stiffeners lead to the same results. The deformed meshes for the full scale and the coupled model are illustrated in Figure 6.6.

The coupled system evidently leads to the correct solution, while at the same time the size of the linear system is reduced considerably. For this completely linear example with small deformations and linear-elastic material behavior, the material matrix $\mathbf{D}$ is constant for every integration point and arbitrary shell strains. It is thus sufficient to determine $\mathbf{D}$ only once for one RVE. In this case, the present coupled method is equivalent to a classical homogenization approach, where constant effective elastic properties of a microstructure are derived by analytical or numerical means, which are then inserted into the homogenized model of the global structure. However, the advantages of the presented method are brought to bear especially in nonlinear systems, whether nonlinear material behavior, nonlinear local geometry, or both are employed.



Elem. x	Full scale	FE2
3	4.424E-01	4.425E-01
6	5.526E-01	5.527E-01
12	5.618E-01	5.619E-01
24	5.641E-01	5.642E-01
36	5.646E-01	5.646E-01
120	5.649E-01	5.649E-01
480	5.650E-01	5.650E-01
analytical	5.650E-01	5.650E-01

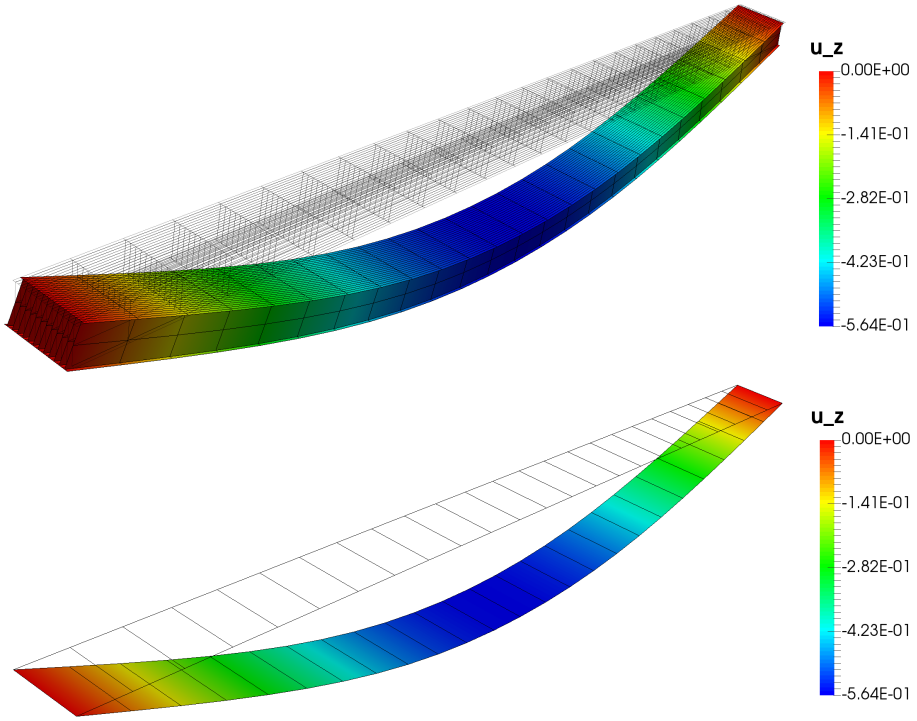


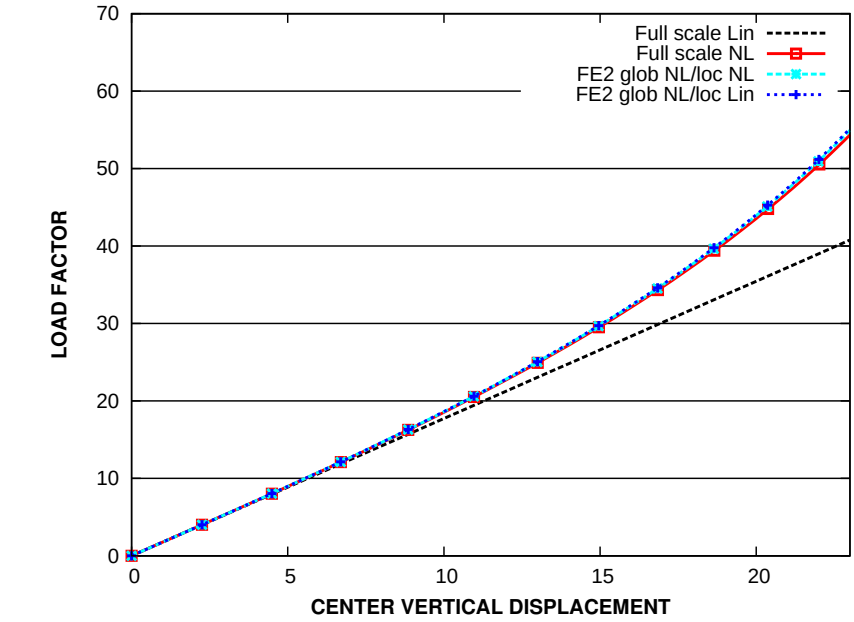
Figure 6.6: (a) Convergence of maximum displacement for the full scale and coupled models. (b) Mesh, deformed mesh (20x amplified) and z-displacement (contour) for the full scale and the coupled model with 24 elements in x-direction.

6.2.3 Geometrically nonlinear deformation behavior

The same system as in the previous section is reconsidered, now assuming finite rotation kinematics instead of linear geometry. It is known that such systems show additional stiffening when subjected to large displacements, as membrane stresses now occur and transfer parts of the load. Once again, a full scale generic shell model with intersections is compared as a reference solution to the coupled model. Now, the linearized stress resultants $\mathbf{D}$ represent the material tangent and, together with the stress resultants $\boldsymbol{\sigma}$, generally depend on the imposed shell strains $\boldsymbol{\varepsilon}$, if nonlinear geometry is employed on the RVE level. Hence, it is required to reevaluate both $\mathbf{D}$ and $\boldsymbol{\sigma}$ in each integration point and at each load step, which would not be possible in a classical homogenization approach. The total load $\lambda \cdot q_z$ is applied in incremental steps using an arc length method [51] as by Riks [80]. Solving the nonlinear model potentially leads to long calculation times for large equation systems. Reducing the size of occurring equation systems with the coupled model, when compared to the fully discretized shell model, is thus especially effective for nonlinear calculations.

Figure 6.7 shows the load-deflection curves for the nonlinear full scale and coupled models, as well as a deformed mesh for the coupled system. The stiffening behavior due to membrane stresses is correctly reflected by both models, leading to a contraction of the system along the x-direction for large displacements u_z. Moreover the system behavior can be accurately reflected by the coupled model regardless of whether linear or nonlinear geometry is assumed on the local scale, as long as nonlinear geometry is enabled on the global scale.

For this particular model, the geometric nonlinearity is only of concern on the global scale, where additional membrane stresses are introduced, changing the load-displacement behavior of the system. On the local scale with an RVE without imperfections, the material tangent $\mathbf{D}$ is actually constant, so that introducing nonlinear geometry only on the global scale would be sufficient. It can be seen in Figure 6.7(a) that the coupled simulation leads to the same results regardless of whether nonlinear geometry is assumed only on the global scale or on both scales. In theory, it would be possible to evaluate $\mathbf{D}$ only once and generate the varying values of $\boldsymbol{\sigma}_i$ by computing $\boldsymbol{\sigma}_i = \mathbf{D}\,\boldsymbol{\varepsilon}_i$ in each integration point i and each load step. This special case is not accounted for in the implementation, as generally the investigation of systems subjected to full geometric (and possibly physical) nonlinearity is more of a concern. Modeling the local scale system with finite rotations can allow to describe effects like stability modes on the local scale, as will be seen in later examples.



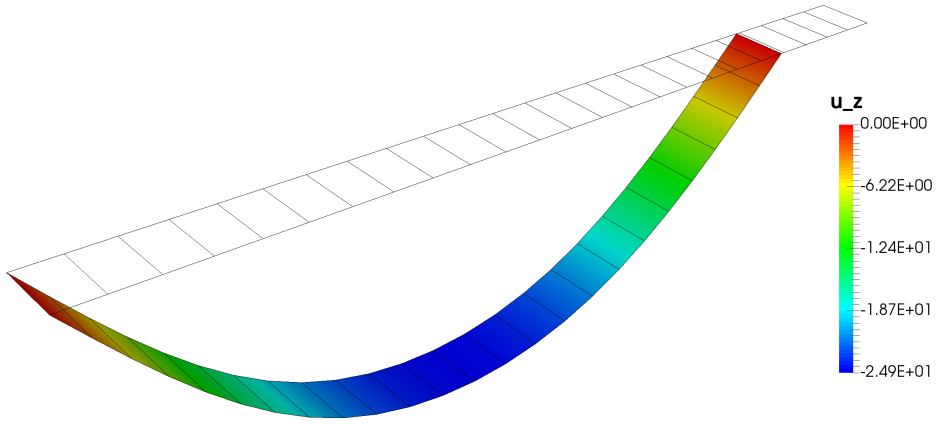


Figure 6.7: (a) Load-deflection curves for the full scale and coupled geometrically nonlinear models. (b) Mesh, deformed mesh (not amplified) and z-displacement (contour) for the coupled nonlinear model ($\lambda = 63.441$).

6.3 Bidirectionally stiffened sandwich cell

As a next system, a sandwich structure comparable to the last section is considered, but now interior stiffeners are placed in both in-plane directions x and y of the shell surface. An RVE will be modeled with n_y^R stiffeners parallel to the x-axis ($y = \text{const}$) and n_x^R stiffeners parallel to the y-axis ($x = \text{const}$), as shown in Figure 6.8. A full system based on this microstructure is illustrated in Figure 6.9.

The grid-like microstructure of this system with cuboid cells can be seen as a first approximation to hexagonal honeycomb cells used in many application cases.

6.3.1 Validation of RVE material matrix

Starting from the RVE of the last section with measurements $l_x^R = l_y^R = 2, h = 3, t = 0.1$, linear-elastic material behavior $E = 7000, \nu = 0$ and $n_y^R = 1$ cell wall along the x-axis, an additional cell wall $n_x^R = 1$ is placed alongside the y-axis. The boundary and link conditions in the finite element model are similar to the unidirectionally stiffened shell as shown in Figure 6.2. Consequently, the material matrix $\mathbf{D}$ largely reflects the structure from Section 6.2.1 as well, with components D_{11}, D_{33}, D_{44} corresponding to (6.3). D_{22} and D_{55} now comprise additional terms from the cell wall(s) $n_x^R > 0$:

$$\begin{aligned} D_{22} &= \frac{EA_x}{l_x} = \frac{E(2 \cdot l_x t + n_x \cdot ht)}{l_x} \\ D_{55} &= \frac{EI_x}{l_x} = \frac{E}{l_x} \cdot \left(\frac{h^2}{2} l_x t + \frac{n_x}{12} h^3 t \right) \end{aligned} \tag{6.9}$$

Due to the choice of parameters $l_x^R = l_y^R$ and $n_x^R = n_y^R$, the RVE behaves in the same way for both directions x and y, thus leading to $D_{11} = D_{22}$, $D_{44} = D_{55}$ and $D_{77} = D_{88}$. Table 6.3 shows the analytically and numerically obtained stiffness components for a coarse and a fine mesh with different choices of parameters h_x^R, h_y^R, h_z^R. It is noted that in the finite element solution very small additional bending components $D_{45} = D_{54}$ are present.

component	D_{11}	D_{22}	D_{33}	D_{44}	D_{55}	D_{66}	D_{77}	D_{88}	D_{45}
analytical	2450	2450	700	3937.5	3937.5	-	-	-	-
FE $2 \times 2 \times 2$	2450	2450	701.74	3938.39	3938.39	1577.31	498.69	498.69	-0.249
FE $64 \times 64 \times 64$	2450	2450	702.14	3938.25	3938.25	1578.76	484.59	484.59	-0.057

Table 6.3: Components of the material matrix $D_{11}, \ldots, D_{88}$ for the bidirectionally stiffened RVE.

With the exception of D_{45}, which is several orders of magnitude smaller than all other values, all stiffness components are very accurate even for the coarse mesh. Where applicable, analytical and numerical results show very good agreement and are even exact

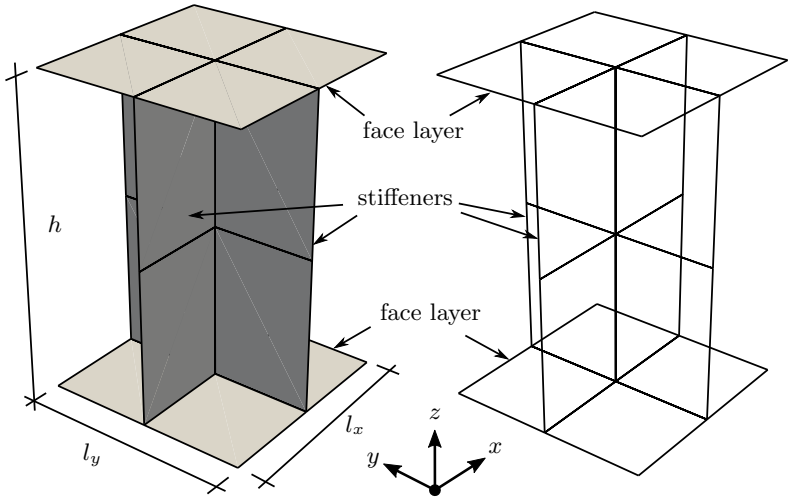


Figure 6.8: Representative volume element with $n_y^R = n_x^R = 1$ cell walls and minimal discretization $h_x^R = h_y^R = h_z^R = 2$ in a solid and wireframe view.

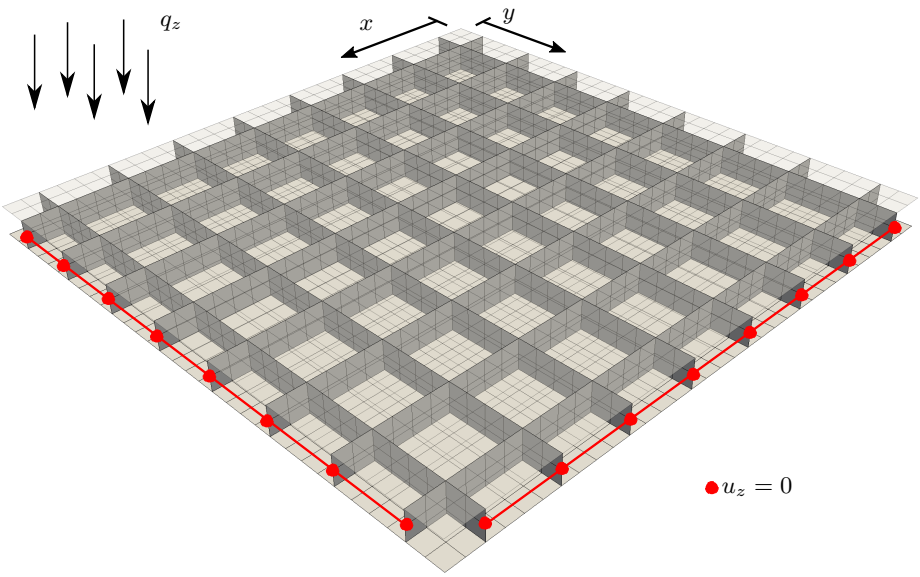


Figure 6.9: Quarter-system (upper face layer drawn semi-transparent) for the bidirectionally stiffened sandwich plate. The full system comprises $n_x = n_y = 16$ cell walls per direction, with discretization parameters $h_x = h_y = 4$, $h_z = 2$. Symmetry boundary conditions on edges $x = 0$, $y = 0$ not shown.

for the membrane components D_{11} and D_{22}. Slight differences in the other components are caused by neglecting terms of higher order in t, for example in the analytic evaluation of I_x and I_y.

Contrary to the previously considered unidirectionally stiffened RVE, all spatial lengths are now predetermined for a given microstructure by the choice of stiffener counts n_x^R and n_y^R in the RVE. Their values lead to the lengths $l_x^R = 2 \cdot n_x^R$ and $l_y^R = 2 \cdot n_y^R$, while the height h and local shell thickness t are fixed. However, the parameters n_x^R and n_y^R can be varied arbitrarily and should lead to the same material matrix $\mathbf{D}$. The influence of n_x^R and n_y^R on the stiffness components is investigated. In fact, only the torsional stiffness D_{66} and the shear components $D_{77} = D_{88}$ vary for the different RVEs.

It can be seen in Figure 6.10 that the torsional stiffness is virtually constant, while the shear components only show a good agreement for small RVEs with $n_x^R = n_y^R = 8$ at most. In applied coupled simulations it is however beneficial to choose RVE sizes as small as possible in order to reduce numerical effort. Moreover, RVEs are by the means of micromechanics required to be small in comparison to the global system size, additionally ruling out the usage of large RVEs. It will be shown in further examples that coupled models using the bidirectionally stiffened RVE lead to correct results despite the rough approximation of shear stiffness components. Similar to the unidirectionally stiffened

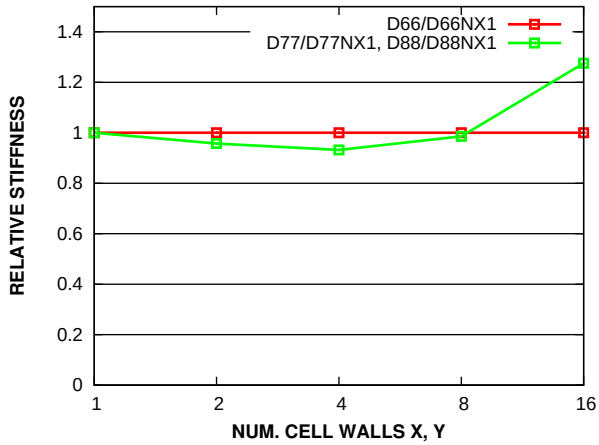


Figure 6.10: Torsional and shear stiffness components for different RVEs with $l_x = l_y = 2 \cdot n_x^R = 2 \cdot n_y^R$, compared to results for $n_x^R = n_y^R = 1$.

RVE, additional terms D_{12} and D_{45} appear on the secondary diagonal for materials with $\nu > 0$. Likewise, membrane, bending and shear modes are still decoupled.

6.3.2 Linear deformation behavior

A quadratic sandwich plate is investigated as the first example with the bidirectionally stiffened RVE under the assumption of linear geometry. The considered microstructure leads to RVEs with $l_x^R = 2 \cdot n_x^R$ and $l_y^R = 2 \cdot n_y^R$ for an arbitrary choice of the number of stiffeners n_x^R and n_y^R within one RVE. On the global scale, $n_x = n_y = 16$ stiffeners are present in the quadratic plate with measurements $l := l_x = l_y = 2 \cdot n_x = 32$ and $h = 0.8$, representing both the height of the RVE as well as the thickness of shell elements on the global scale of the coupled model. $t = 0.1$ represents the shell thickness of stiffeners and face layers on the local scale. The mesh with outer boundary conditions and load is illustrated in Figure 6.9.

The structure is subjected to a constant area load $q_z = 0.001$ with material data $E = 7000, \nu = 0.34$. On its boundaries $x, y = \pm l/2$ a Navier-type soft support is assumed where only displacements u_z are fixed. As the resulting system has both x and y as symmetry axes, only a quarter of the global structure is discretized with $x, y \in [0, l/2]$. Symmetry boundary conditions $u_x = \omega_y = 0$ and $u_y = \omega_x = 0$ are imposed on the boundaries $x = 0$ and $y = 0$, respectively, where ω_x and ω_y denote the rotational degrees of freedom with respect to the global x- and y-axes. Due to the choice of boundary conditions, no additional nodes have to be fixed to prevent rigid body motions.

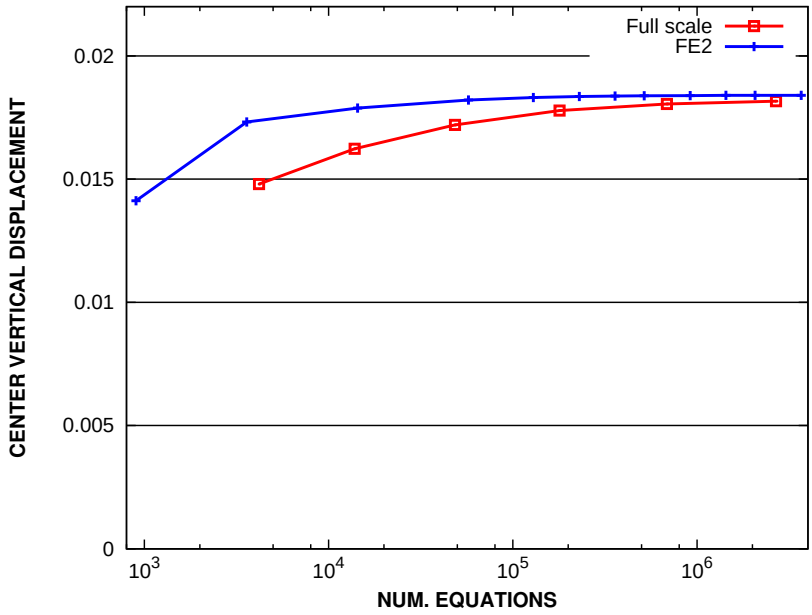
As a first very rough approximation the analytical result for the maximum displacement of a homogeneous quadratic plate with side length l, Navier boundaries, and a constant area load q_z is considered:

$$w_{\max} = 0.004055 \, \frac{q_z l^4}{K} \quad \text{with} \quad K = \frac{E h^3}{12(1-\nu^2)} \tag{6.10}$$

For a homogeneous plate the parameter K is represented by the bending stiffness components D_{44} and D_{55}. The RVE corresponding to this example leads to parameters $D_{44} = D_{55} = 269.8851$, so that with equation (6.10) the approximation for a homogeneous plate with the same plate stiffness follows: $w_{\max,\text{hom}} \approx 0.01575$

In order to minimize calculation times for the numerical simulation, the smallest possible RVE discretization with $n_x^R = n_y^R = 1$ and two elements per side is chosen ($h_x^R = h_y^R = h_z^R = 2$), as shown in Figure 6.8. The solution obtained from the coupled model is compared to a full scale reference solution for varying numbers of elements on the global scale.

Results are plotted in Figure 6.11(a), where the maximum displacement is shown with



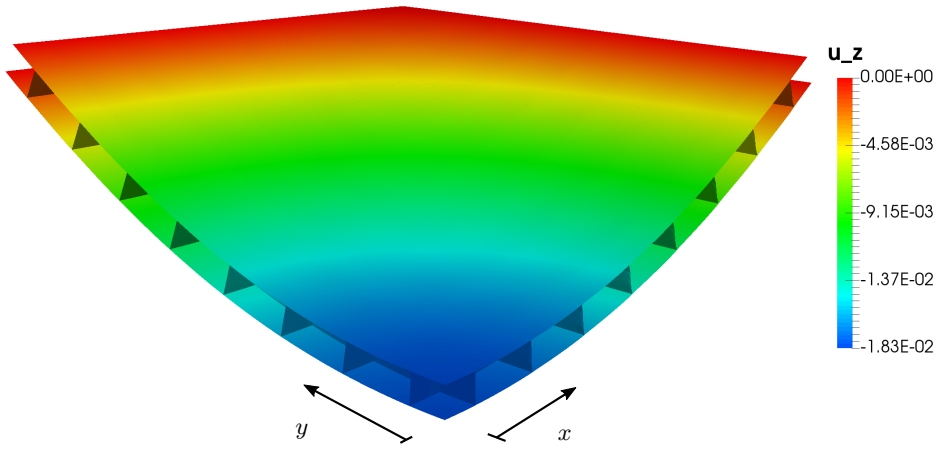


Figure 6.11: (a) Convergence of maximum displacement for the full and coupled finite element models. (b) Deformed mesh (300x amplified) and z-displacement (contour) for the full model.

respect to the total number of equations for both the full scale model and the coupled system. The numerical results for the sandwich plate converge to a maximum displacement value of $w_{\max} = 0.01840$ and differ by roughly 17% from the analytical approximation for a homogeneous plate $w_{\max,\text{hom}}$. In order to determine the number of equations for the coupled model, the number of equations for one RVE neq_{rve} was multiplied with the number of integration points per shell element on the global scale, $ngp = 4$, and the number of shell elements on the global scale n. This total number of local scale equations neq_{loc} was then added to the number of global scale equations neq_{glob}, resulting in the final number of equations neq.

$$neq = neq_{glob} + neq_{loc} = neq_{glob} + n \cdot ngp \cdot neq_{rve} \tag{6.11}$$

In this example, the chosen RVE discretization with one stiffener in each direction x, y and two elements per side leads to $neq_{rve} = 223$. Thus, a model with 32×32 shell elements on the global scale ($neq_{glob} = 5248$) would result in $neq = 5248 + 32^2 \cdot 4 \cdot 223 = 918656$ total equations. The data points in Figure 6.11(a) actually represent FE2 systems with $1, 2, \ldots, 64$ shell elements per side in the global system ($n = 1^2, 2^2, \ldots, 64^2$).

When comparing a full scale model and a coupled model with the same number of equations, the coupled model not only yields a better converged result, it also calculates much faster. This is due to two reasons;

a) all local equation systems are independent from each other, thus most of the FE2 system's equations are in fact decoupled,

b) most importantly, in the linear case, all RVEs lead to the same linearized stress resultants, making it sufficient to evaluate the local system only once.

In fact, for the largest investigated coupled model (64×64 global elements, $neq = 3674368$), a solution time (assemble stiffness matrix, triangular decomposition, solution of equations) of only 1.5 seconds was observed, while the largest full scale model ($neq = 2684578$) required 350 seconds to compute on the same machine.

It is noted that this extreme speedup factor can not be preserved in the nonlinear calculations where each RVE must be evaluated individually in every iteration and load step. However, the coupled method still leads to smaller and partially decoupled system matrices, allowing for a quicker solution procedure when compared to the full scale model.

6.3.3 Comparison of stress resultants

In order to show that not only the global deformation behavior but also the local state of the full scale system can be reflected by the coupled simulation, a closer look is taken at the occurring stress resultants in a deformed system. It would be expected that, wherever applicable, along a vertical line (x^*, y^*, z), with x^* and y^* fixed and $z \in [-h/2, h/2]$, the stress resultant curve through the height of a cell wall in the full scale model corresponds to the stress resultant curve through the cell wall in an RVE in a Gauss point (x^*, y^*) of the coupled model. However, an exact match where (x^*, y^*) represents at the same time the location of a Gauss point, as well as the location where a cell wall is placed, can usually not be achieved. Moreover, the stress state within the RVE is evidently not homogeneous so that the exact curves in the RVE depend on the location in the local system at which the stress resultants are evaluated. Thus, the RVE stress resultant curves can only be understood as a first approximation of the local state in the full scale model.

For the bidirectionally stiffened sandwich plate, a point near the symmetry corner $A \approx (0, 0)$ lends itself to closer investigation. In addition to that, a second point close to the center of the discretized quarter-plate $B \approx (l/4, l/4)$ is chosen. The stress resultants of the full scale model are evaluated in the nearest cell wall in either direction, using the geometry described in the previous section, while for the coupled simulation the local stress resultants for the RVE corresponding to the closest integration point are investigated. An RVE with $n_x^R = n_y^R = 4$ cell walls per direction is used along with a fine discretization both in the full scale system and in the coupled system, where $h_x = h_y = h_z = 8$ elements are present between two cell walls in both directions, as well as between the face layers in thickness direction.

Element stress resultants are plotted only for the vertically aligned cell wall elements, to avoid any influences from the face layers. For this reason, no values are obtained for the upper- and lowermost ends of the cell walls where $z = \pm h/2 = \pm 0.4$, but only for the interval between the outermost Gauss points $z \in [-0.38, 0.38]$. All coordinate directions are transformed to the global scale (x, y, z) coordinates, so that for a cell wall with $y = \text{const}$ (parallel to the x-axis), x and z actually reflect the in-plane coordinates of the cell wall. To be able to distinguish stress resultants referring to the global or local coordinates, the notation

$$\begin{aligned} N_x &:= n^{11} \quad \text{(Normal in-plane force in length direction of cell wall)} \\ N_z &:= n^{22} \quad \text{(Normal force in global height direction)} \\ N_{xz} &:= n^{12} \quad \text{(Shear forces along length/height directions)} \end{aligned} \tag{6.12}$$

$$
\begin{aligned}
M_x &:= m^{11} \quad \text{(Bending moment w.r.t. length axis of cell wall)} \\
M_z &:= m^{22} \quad \text{(Bending moment w.r.t. global height axis)} \\
M_{xz} &:= m^{12} \quad \text{(Cell wall torsional moment)} \\
Q_x &:= q^1 \quad \text{(Transversal shear force at lateral boundaries)} \\
Q_z &:= q^2 \quad \text{(Transversal shear force at upper/lower boundaries)}
\end{aligned}
\tag{6.13}
$$

is introduced, where $n^{\alpha\beta}$, $m^{\alpha\beta}$, q^{α} correspond to the local cell wall axes and were introduced in (3.24). $N_x, \ldots, Q_z$ refer to the global coordinate system instead.

The stress resultants are obtained as integrals of local stresses over the cell wall thickness, so that normal and shear resultants $N_{\alpha\beta}$, Q_α are given as force per length $[kN/cm]$, while the resulting moments $M_{\alpha\beta}$ must be understood as moment per length $[kN \cdot cm/cm]$. Due to the symmetry of the system, it is sufficient to only evaluate the local state along either one of the cell wall directions $y = \text{const}$ or $x = \text{const}$.

The stress resultant curves are compiled in Figure 6.12. It can be seen that for the point A near the symmetry corner, only the in-plane normal force N_x is relevant, which can be perfectly described by the coupled model when compared to the full scale solution. The normal force in thickness direction N_z can only roughly be reproduced, however, N_z is two orders of magnitude smaller than N_x and does not contribute significantly to the global behavior. All other stress resultant components are small in A when compared to the point B at the center of the quarter-plate.

In B, the behavior of the normal forces corresponds to the point A: The dominant in-plane normal force N_x can accurately be described, the thickness normal force N_z is only moderately accurate but small. The bending and torsional moments M_x, M_z and M_{xz} agree well between the coupled and the full scale simulations. For N_{xz} and Q_x, which are one order of magnitude smaller than N_x, the results coincide reasonably well. The only significant discrepancies between both models occur in the shear force Q_z, which is acceptable due to Q_z being significantly smaller than the dominant stress resultant forces and especially N_x.

Given the aforementioned difficulties of comparison between two models without a clear spatial or physical correlation between cell wall elements in a full scale model and cell wall elements within an RVE, the description of local quantities in the coupled model is all in all satisfyingly accurate.

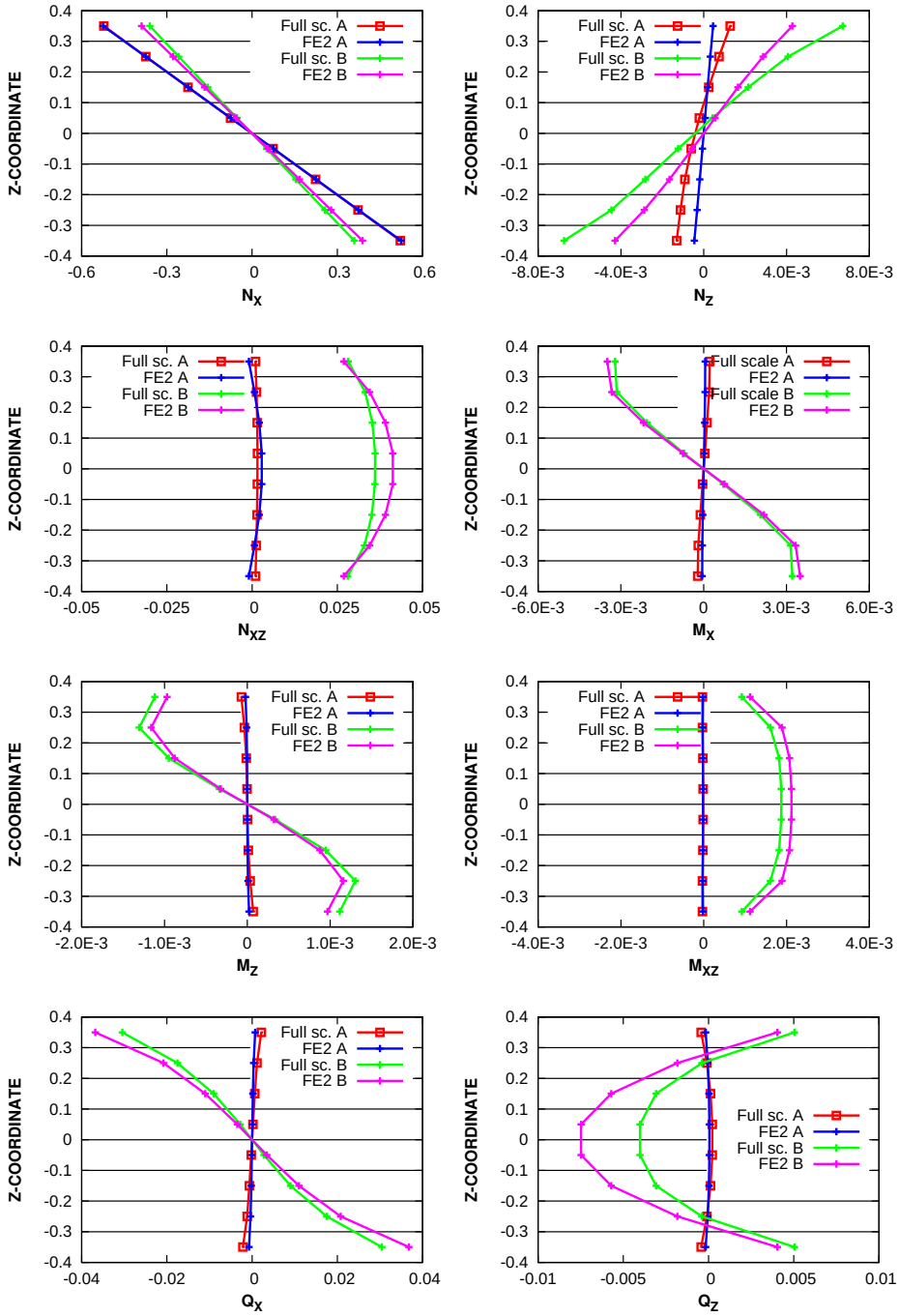


Figure 6.12: Stress resultant curves $N_x, \ldots, Q_z$ for points near the symmetry corner $A \approx (0, 0)$, and near the center of the quarter-plate $B \approx (l_x/4, l_y/4)$, plotted through the thickness direction for full scale and coupled simulations. Resultant forces $N_{\alpha\beta}, Q_\alpha$ are given in $[kN/cm]$ while moments $M_{\alpha\beta}$ must be understood as $[kN\,cm/cm]$.

6.3.4 Geometrically nonlinear deformation behavior

All of the examples covered up until now featured completely linear behavior on the local scale, so that a linear-elastic material law and small local rotations could be assumed. In these cases, the coupled global-local method corresponds to a classical homogenization method as was mentioned before. In order to fully utilize the new possibilities of the proposed method, nonlinear behavior on the local scale must be introduced, possibly in terms of a nonlinear material law, nonlinear geometry, or both. This section will cover an example regarding nonlinear geometry, where face sheet buckling will arise due to using an RVE with geometric imperfections. In contrast to this, in the next section a nonlinear elasto-plastic material law will be introduced for modeling physically nonlinear behavior.

A system similar to the one illustrated in Figure 6.9 will be used as a starting point, i.e. a doubly symmetric sandwich plate with Navier-type boundary conditions on the outer edges, loaded by a scaled constant area load $\lambda \cdot q_z$ with $q_z = 0.01\,[kN/cm^2]$. The global measurements are chosen as $l_x = l_y = 32$ and $h = 0.8$, with cell walls placed in regular intervals of $2\,[cm]$, so that $n_x = n_y = 16$ cell walls per side are placed in total. Shell thicknesses $t = 0.1$ are used for the cell walls and the face layers. For the global scale model, a discretization of 4×4 5-DOF shell elements is employed along with symmetry boundary conditions on the x- and y-axes. The possible occurrence of non-symmetric buckling modes is thus eliminated from this simulation. An RVE is chosen for the local model with $n_x^R = n_y^R = 2$ cell walls in each direction and discretization parameters $h_x^R = h_y^R = 16$, $h_z^R = 2$. The RVE thus comprises one fully closed cell in its center, as shown in Figure 6.13. A geometric imperfection is introduced into the RVE by translating the center node in the upper face layer, which has the local coordinates $(0, 0, h/2)$ in the perfect system, by a vertical offset $\Delta z = h/200 = 0.004\,[cm]$. All other nodes within the inner region $|x| < l_x^R/4$, $|y| < l_y^R/4$ of the upper face layer are relocated accordingly, so that a cupola-shaped dent is set up between the cell walls.

In the coupled simulation, linear geometry is assumed for the global model, but the RVE is calculated with the assumption of finite rotations to allow for local buckling. The nonlinear calculation is carried out using an arc length method, leading to the load-displacement behavior shown in Figure 6.15(a). It can be seen how the geometrically nonlinear model decreases in system stiffness after passing a bifurcation point (approximately at $\lambda = 7$). The geometric imperfection in the RVE leads to the occurrence of local face sheet buckling as shown in Figure 6.14, where the deformation of an RVE, near the center of the structure, at the end of the calculation ($\lambda = 20$) is illustrated.

Evidently, the nonlinear geometry in the local system leads to a non-constant matrix of

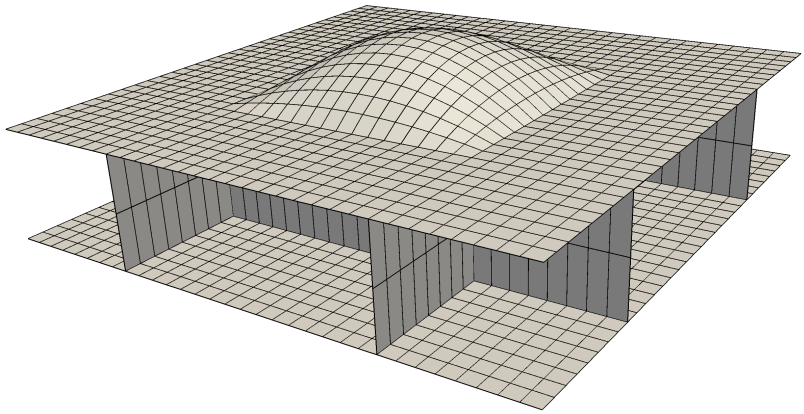


Figure 6.13: Exemplary RVE with $n_x^R = n_y^R = 2$ cell walls per side and geometric imperfection (amplified 100×) in the upper face layer.

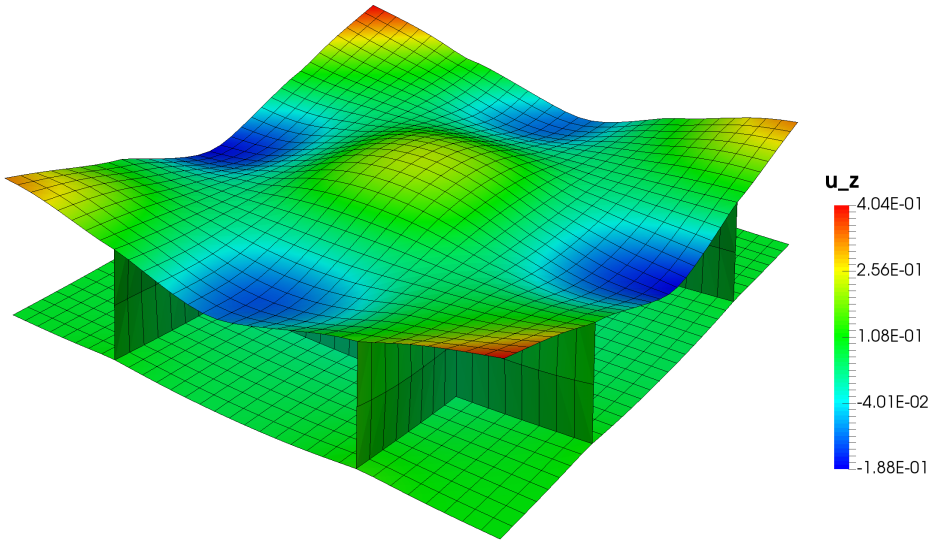


Figure 6.14: Deformation (not amplified) and vertical displacement (contour) for an RVE near the center of the global structure with $\lambda = 20$.

linearized stress resultants, depending on the prescribed displacements, so that it is no longer possible to only solve the local equation system once. Instead, one local boundary value problem must be solved in every integration point of all global shell elements. This leads to higher calculation times on the one hand, but on the other hand the coupled model is well able to describe the occurrence of buckling as well as the post-critical system behavior, unlike approaches such as the classical homogenization method.

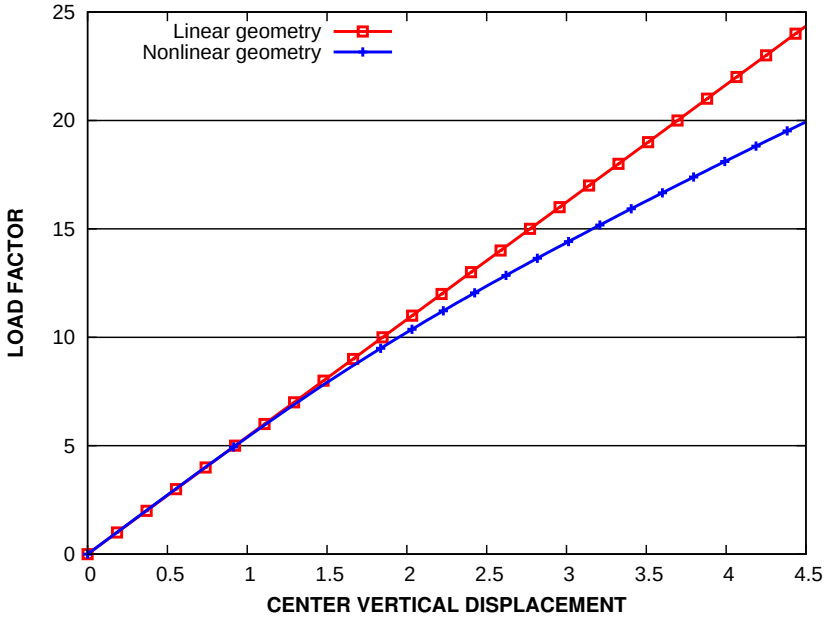


Figure 6.15: Load-deflection curve for linear and nonlinear geometry in the local system.

6.3.5 Physically nonlinear deformation behavior

In the previous examples, only linear-elastic material behavior was investigated. As a first step for nonlinear material laws, a small strains elasto-plastic material [67, 75], also shown in [84], is considered.

Starting from the same model for the bidirectionally stiffened plate as specified in Section 6.3.2 and just replacing the material law, it turns out that the boundary conditions and geometric measurements of the system lead to problems with nonlinear material behavior: As the structure is supported only in nodes on its middle surface $z = 0$, in fact only a single node at the end of each stiffener is supported as a boundary condition - see Figure 6.16(a). This leads to high reaction forces in the supported nodes and thus to a localization of plastic flow in the stiffeners at the boundaries of the system, as shown in Figure 6.16(b). In the coupled simulation, this effect can not be described, as elasto-plastic behavior only occurs on the local scale where the boundary conditions of the global system are not directly visible. Attempts of a comparison between the two elasto-plastic systems do thus not lead to agreeing results, the coupled model is not able to describe effects localized at the very boundary of the global structure.

With the existing boundary conditions and geometry turning out to be unsuitable for the description of physical nonlinearity, other structures will be considered as examples for modeling nonlinear material behavior in coupled simulations. It will be seen that outside of the proximity of the boundaries, elasto-plastic material behavior can be accurately described by the coupled model. Three different examples will be investigated, where the hitherto system is slightly modified in order to let the initial occurrence of plastic flow localize in the inner parts of the structure or at the symmetry boundaries.

For all examples, the boundary conditions are revised to lock the vertical displacements u_z at the edges $x, y = l/2$ not in the mid-surface $z = 0$, but throughout the full thickness of the structure instead, leading to lower per-node reaction forces due to a higher total number of supported nodes. With these boundary conditions, the applied shear forces are transferred over the full height of the stiffeners instead of a single node and also enter into the face layers.

Additionally, for the first example, the local shell thickness of the stiffeners t_{cw} is increased when compared to the thickness of the face layers t_{fl}, in order to prevent shear force failure in the stiffeners. While on one hand, in a real-world sandwich structure the cell wall material is usually thinner and less stiff than the face layers, on the other hand the cells would be much finer, leading to more stiffeners per unit area of the structure for an overall comparable volume fraction for the cell walls.

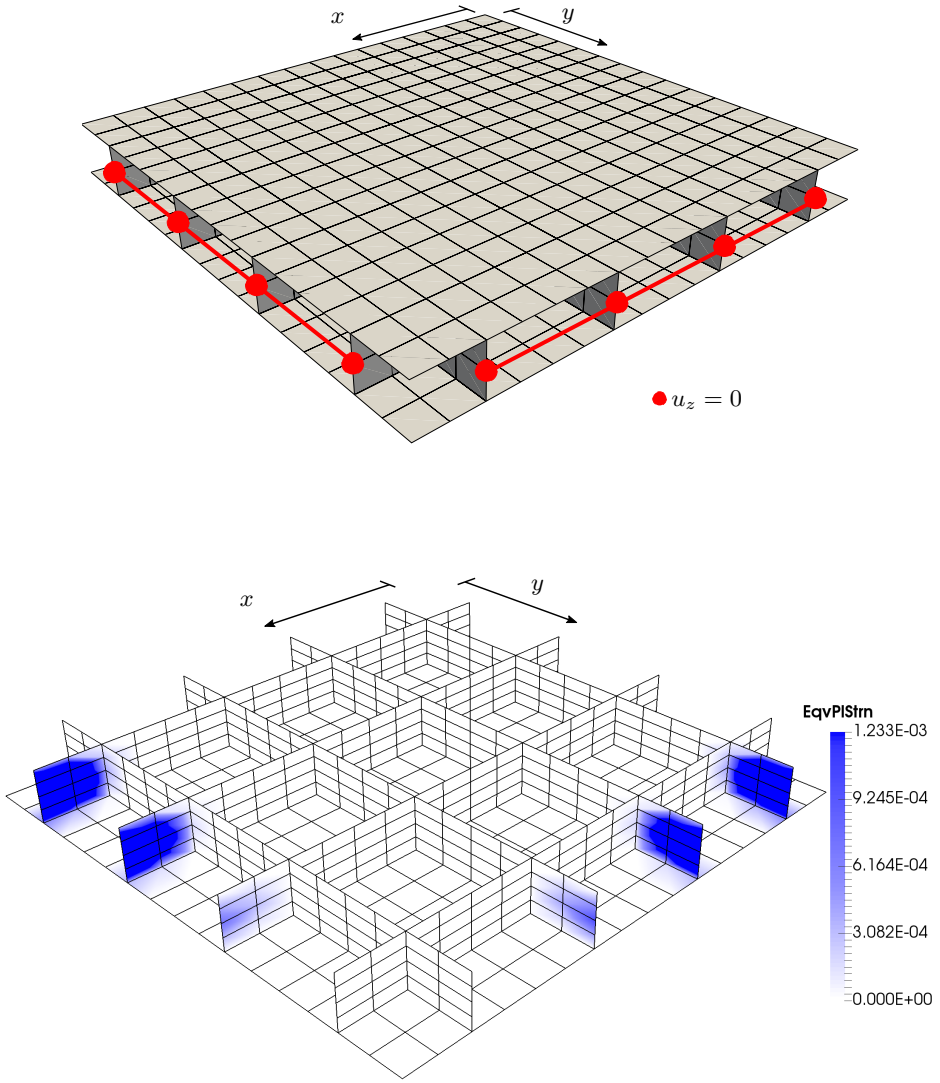


Figure 6.16: (a) Supported nodes and (b) equivalent plastic strain localized at the supported nodes in a sample full scale quarter-plate with $n_x/2 = n_y/2 = 4$ stiffeners.

In the second strategy, the boundary regions of the global structure are modeled with linear-elastic material behavior in order to prevent localization of plastic flow at the supported nodes. This model leads in fact to a better agreement between the full scale and the coupled simulation, as the coupled model is inherently unable to reflect local effects at the very boundaries of a global structure, where no integration points are present.

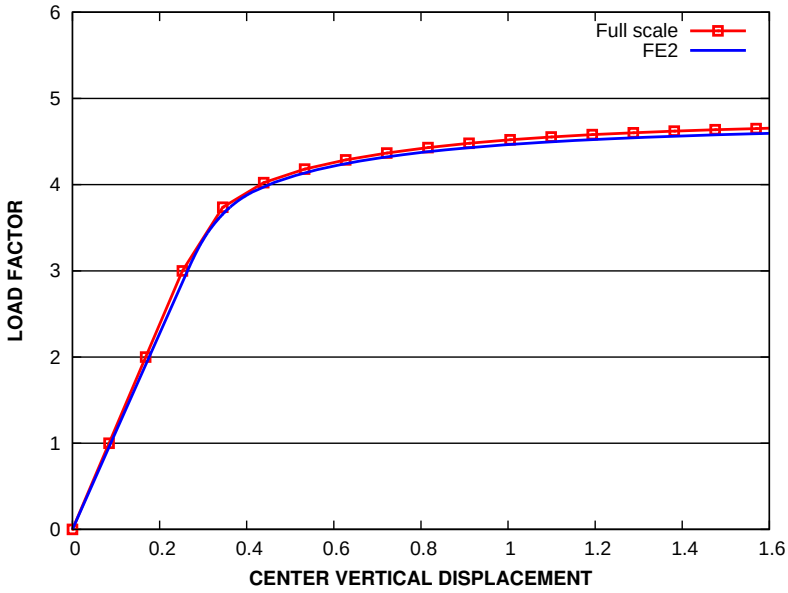
As a third case, an attempt is made to model a more practice-oriented sandwich with a much finer cell structure. The distance between two adjacent cell walls is reduced from $2\,[cm]$ to $0.25\,[cm]$, leading to a significantly larger equation system. In order to compensate for the increased amount of cell walls, the cell wall thickness is reduced to $t_{cw} = 0.025\,[cm]$, down from the previous choice of parameters of $0.1\,[cm]$.

a. Thicker stiffeners

Starting from the linear system out of Section 6.3.2, the thickness of the stiffeners is increased to $t_{cw} = 0.3$, while the face layers are kept at $t_{fl} = 0.1$. Additionally, the boundary conditions on the outer edges are adjusted in the full scale model, so that $u_z = 0$ on all nodes with $x = l_x/2$ or $y = l_y/2$, dropping the former additional requirement $z = 0$. It is noted that with the assumption of small rotations ($\cos\beta_\alpha \approx 1$), these conditions do not inhibit rotations around the boundary axis at the boundary edge. In the coupled model, no further modifications are necessary for the boundary conditions.

As a material law, von Mises plasticity is used with material data $E = 7000, \nu = 0.34$, initial yield stress $Y_0 = 10$ and no hardening modulus. The system is subjected to a scaled constant area load $\lambda \cdot q_z$ with $q_z = 0.005$. All other parameters remain as in Section 6.3.2, so that $l_x = l_y = 32$ with $n_x = n_y = 16$ stiffeners and $h = 0.8$. Similar to the geometrically nonlinear calculation in Section 6.2.3, a nonlinear simulation based on the arc length method is carried out for both the full and the coupled model, in order to be able to cope with very small load increments for growing displacements.

Figure 6.17(a) shows the load-displacement curves for the coupled model and the full scale solution. The coupled simulation is able to closely reflect the behavior of the full model. Moreover, it can be seen in the plot of plastic equivalent strain in Figure 6.17(b) that the occurrence of plastic flow is now concentrated at the center of the plate, especially in the face layers, instead of the cell walls at the supported boundaries. For this simulation four layers with two gauss points each were used and the equivalent plastic strain is plotted for the outermost integration points. It is noted that the assumption of small strains is easily satisfied.



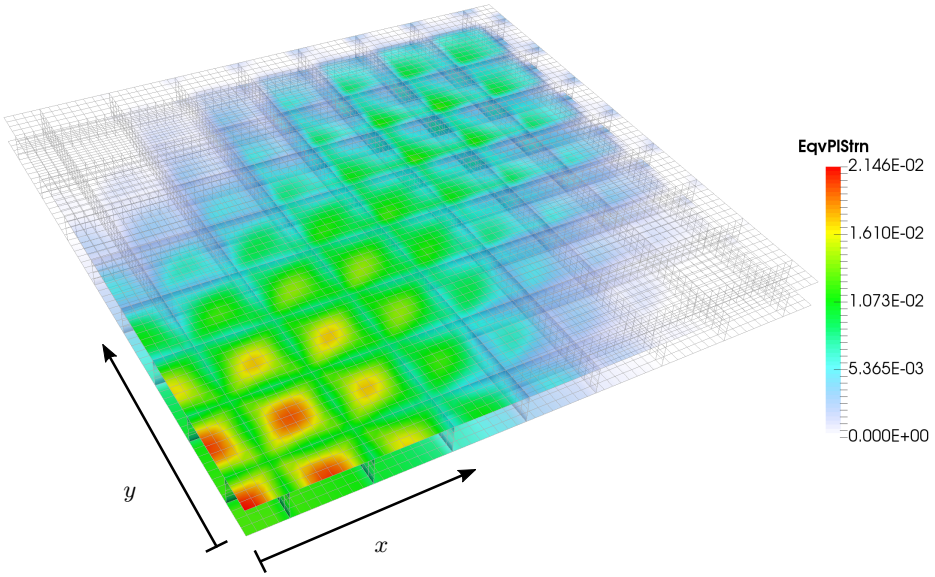


Figure 6.17: (a) Convergence of maximum displacement for the full and coupled finite element models. (b) Equivalent plastic strain concentrated in the upper face layer near the structure's center.

In addition to comparing the global deformation behavior, the local stress state is compared between the full scale and the coupled nonlinear models, just as it was done in Section 6.3.3 for the linear case. The stress state is evaluated near the end of the nonlinear simulation, where $\lambda = 4.649$ (full scale) or 4.6014 (FE2) and $w_{\max} = 1.6691$ or 1.6441, respectively. For the investigation, a point near the center of the quarter-plate, in the stiffener with $y = 7 = \text{const}$ and with $x \approx 7$ was considered. The stress resultant curves are plotted through the thickness direction in Figure 6.18. Once again, N_x is by far the dominant component, being at least one order of magnitude larger than the other stress resultant components. N_x also shows very good agreement between the full scale and coupled simulation, whereas the less significant components can be described with moderate accuracy. Despite the multiple sources of abstraction (no direct spatial correlation between cell walls, points of investigation not matching exactly, total load factor not matching exactly) the stress state can be reflected reasonably well by the RVE.

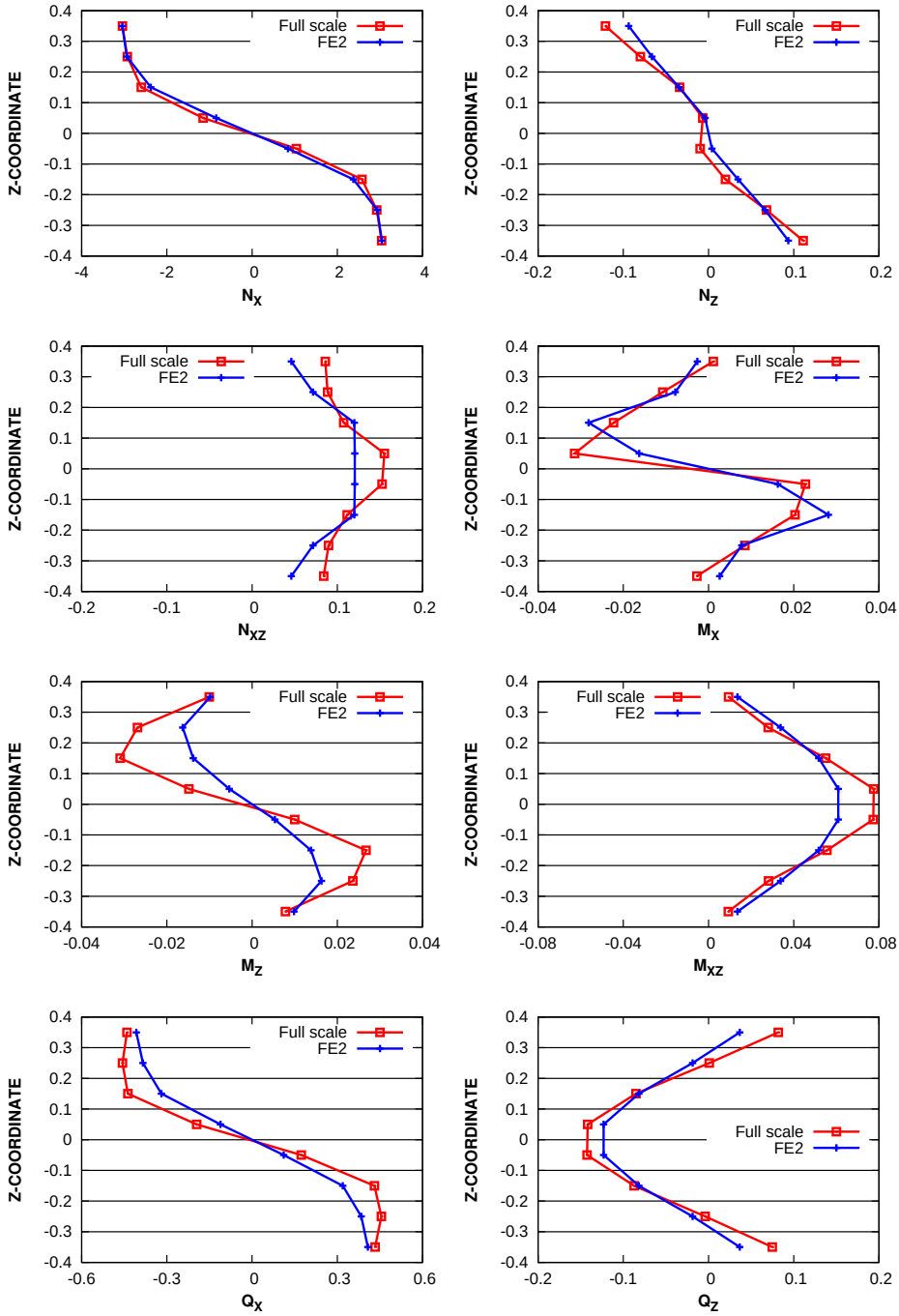


Figure 6.18: Stress resultant curves $N_x, \ldots, Q_z$ near the center of the quarter-plate $(l_x/4, l_y/4)$, plotted through the thickness direction for full scale and coupled simulations. Resultant forces $N_{\alpha\beta}, Q_\alpha$ are given in $[kN/cm]$ while moments $M_{\alpha\beta}$ have to be understood as $[kN\,cm/cm]$.

b. Two-material modeling

Figure 6.19 shows the discretization of an exemplary quarter-plate using the two-material strategy, where symmetry boundary conditions are imposed on boundaries $x = 0$, $y = 0$ and a Navier-type soft support ($u_z = 0$) is used on the outer boundaries $x = l_x/2$, $y = l_y/2$, just as in the previous example. The elasto-plastic material law is used in the inner subdomain

$$\Omega_{pl} = \left[0, \frac{3}{4} \cdot l_x/2\right] \times \left[0, \frac{3}{4} \cdot l_y/2\right] \times [-h/2, h/2] \ , \tag{6.14}$$

while in the outer subdomain the linear-elastic material law is employed:

$$\Omega_{el} = \left]\frac{3}{4} \cdot l_x/2, l_x/2\right] \times \left]\frac{3}{4} \cdot l_y/2, l_y/2\right] \times [-h/2, h/2] \tag{6.15}$$

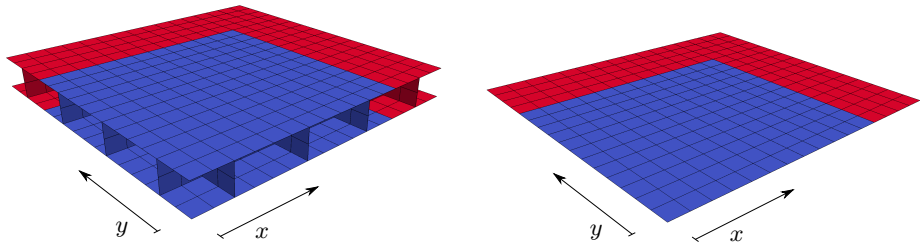
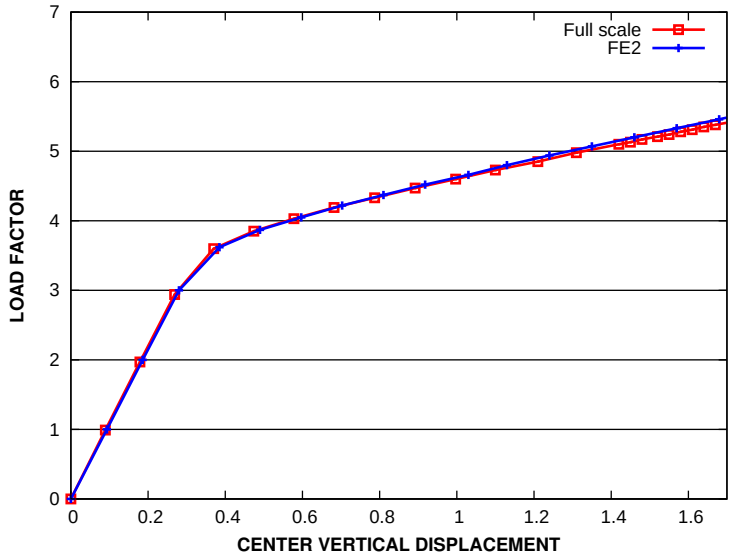


Figure 6.19: (a) Full scale discretization: Elasto-plastic (blue) and linear-elastic material (red) material.
(b) FE2 discretization: Elasto-plastic (blue) and linear-elastic (red) material.

The number of stiffeners is chosen as $n_x = n_y = 16$ so that $l_x = l_y = 32$ are given as global in-plane lengths. All model parameters apart from the two different material laws are chosen as in Section 6.3.2, i.e. height $h = 0.8$, shell thickness $t = 0.1$ for the full model and the RVEs. Material data for the linear-elastic material is the same as before ($E = 7000, \nu = 0.34$) while the initial yield stress for the elasto-plastic material is set to $Y_0 = 10$. The constant area load is chosen as $q_z = 0.005$ and scaled with a load factor λ for a total load of $\lambda \cdot q_z$. Linear geometry was assumed on the global and the local scale. The inner elements with elasto-plastic material behavior are modeled with four layers and 2 gauss points, respectively.

Figure 6.20(a) shows the load-displacement curves for both systems, which agree very well. As was to be expected, the systems behave linearly-elastic for small loads and drop



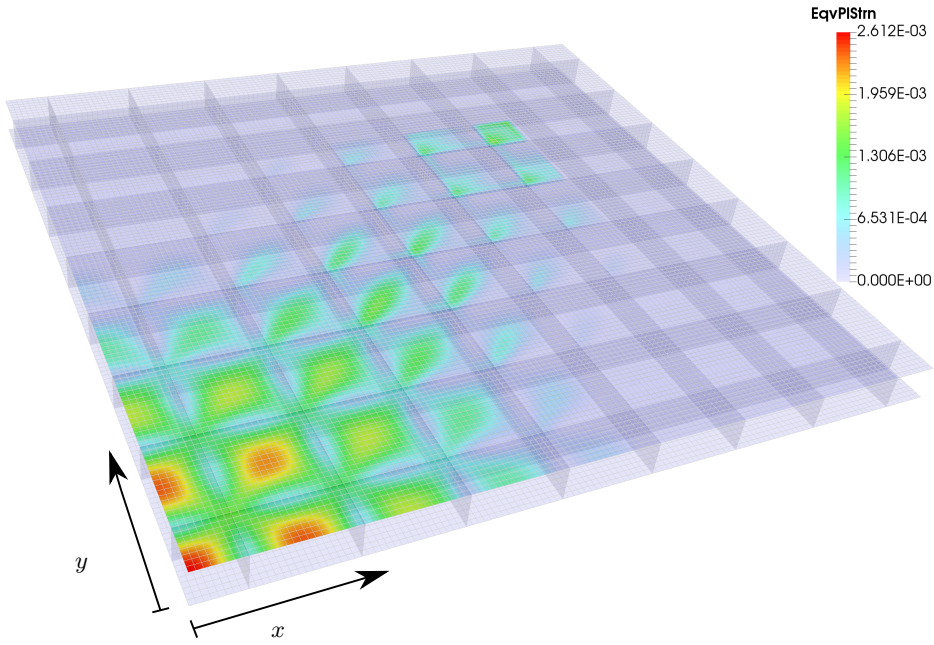


Figure 6.20: (a) Convergence of maximum displacement for the full and coupled finite element models. (b) Equivalent plastic strain in the full scale model (outermost of 4 layers/2 gauss points).

in stiffness after the initial yield stress has been reached by imposing a sufficiently large load.

The resulting equivalent plastic strain for the first load step in which plastic flow occurs ($\lambda = 3.54$) can be seen in Figure 6.20(b). In this system, the localization of plastic strain now appears first in the upper face layer at the symmetry corner, spreading from there to the outer edges of the system. The behavior roughly corresponds to that already shown in Figure 6.17(b) when using thicker cell walls.

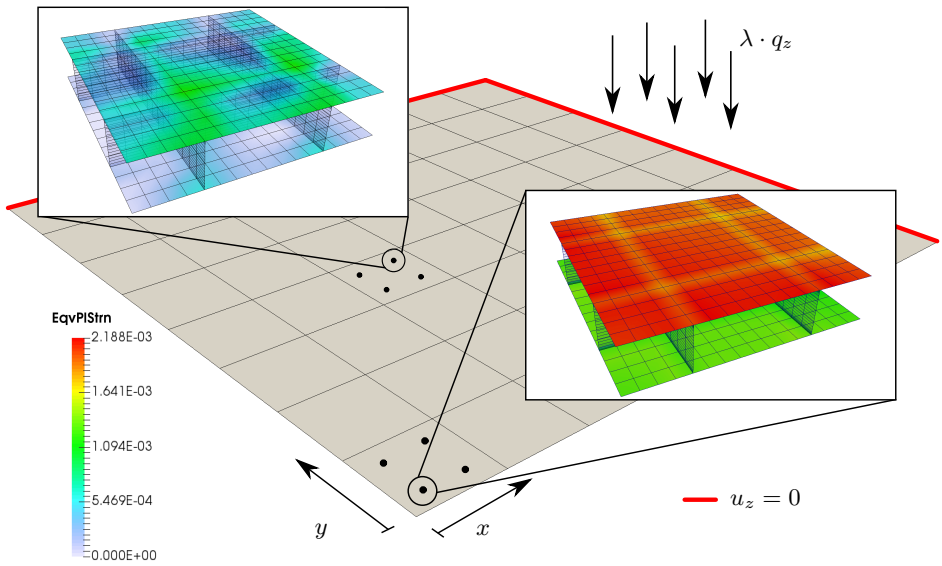


Figure 6.21: Equivalent plastic strain (outermost integration point) in two selected RVEs of the coupled model, one near the symmetry corner $(0, 0)$, and the second more towards the center of the system.

In Figure 6.21 it is shown how this effect can be visualized by means of the coupled model. The global scale model of the coupled system is drawn in gray. For two specific elements of the global model their four integration points have been marked. One of each is chosen and the associated RVE is investigated in the zoomed-in views, where the equivalent plastic strains are plotted as contour color and opacity. It can be seen how the cell walls are completely transparent and thus not plastically deformed. For the first chosen element near the symmetry corner, the plastic strains in the upper face layer are very large, just as in the full scale model in Figure 6.20(b). However, the maximum value is still lower than in the full scale model as the chosen integration point is not located directly at the symmetry corner. In the second point further to the center the plastic strain still concentrates on the upper face layer, but to a much lesser extent.

c. Large scale simulation with many cells per unit area

The examples considered until now have been of small dimensions, in order to be able to generate results more quickly. However, they can not show the full capabilities of the coupled algorithm, as computational overhead overshadows the theoretical performance advantages and leads to higher computation times when compared to the full scale systems, except for linear calculations where only one RVE has to be evaluated once.

In addition, the thickness of the face layers matched the cell wall thickness for the sake of simplicity, or was chosen even thinner than the cell walls in order to prevent plastic flow at supported boundary nodes in 6.3.5(a). In reality, however, a typical sandwich panel would rather consist of very thin cell walls, forming cells with small lateral dimensions. The face layers would be comparably thicker and particularly contribute to the panel's bending stiffness.

Hence, as a more practice-oriented example, a structure is modeled based on the one in Section 6.3.2, but with rectangular cells with in-plane lengths of only $l_x = l_y = 0.25\,[cm]$, leading to a drastically increased number of cell walls and thus an increased number of equations in the full scale system. To compensate for the higher number of cell walls, the cell wall thickness is scaled to $t_{cw} = 0.025\,[cm]$, down from 0.1 or $0.3\,[cm]$ in the previous systems. All other parameters are chosen as before, such as the global measurements l_x, l_y, h and material data E, ν, Y_0. The resulting *minimal* discretization of the full scale quarter-plate is illustrated in Figure 6.22. This model with two elements between cell walls and two elements through the thickness already leads to an equation system with 255616 equations, a number quickly increasing for finer discretizations. The largest full scale model that could still be computed with FEAP comprised $h_x = h_y = h_z = 6$ elements between each cell wall as well as in thickness direction, for a total of 2716800 equations.

On the other hand, the coupled equation system does not increase in size, as the RVEs are just spatially smaller than before, provided the number of cell walls $n_x^R \times n_y^R$ within one RVE is kept constant. A full scale simulation with $h_x = h_y = h_z = 6$ and a coupled simulation with 12×12 global scale elements and RVEs with $h_x^R = h_y^R = 8$, $h_z^R = 4$ and $n_x^R = n_y^R = 1$ are carried out. The results of the simulations are compared in Figure 6.23 and agree very well. Moreover, for this system the coupled model requires much less calculation time when compared to the full scale solution, even despite the implementation overhead. It comes as no surprise that the performance advantage of the coupled method increases for very fine cell structures, which happen to be prevalent in many cases of practical sandwich construction.

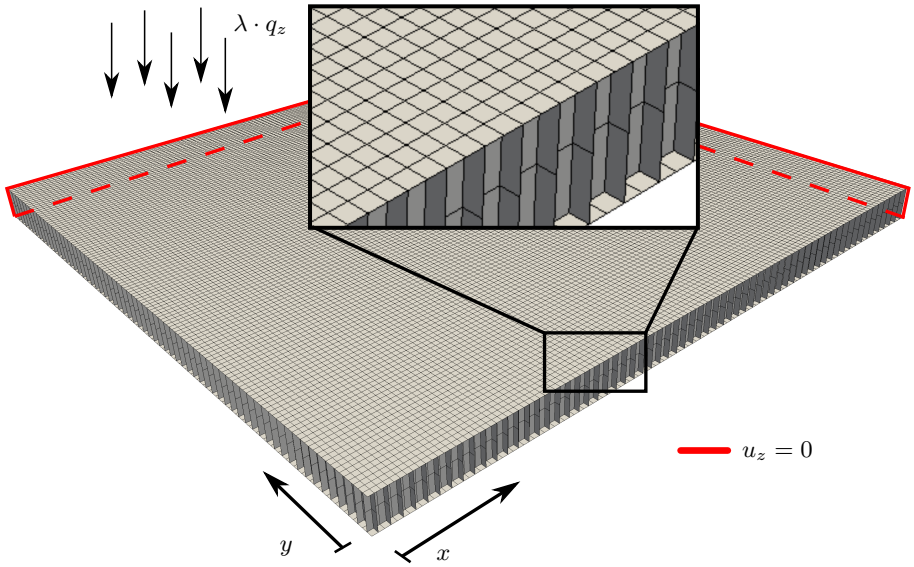


Figure 6.22: Full scale mesh of the quarter-plate in the most coarse discretization ($neq = 255616$).

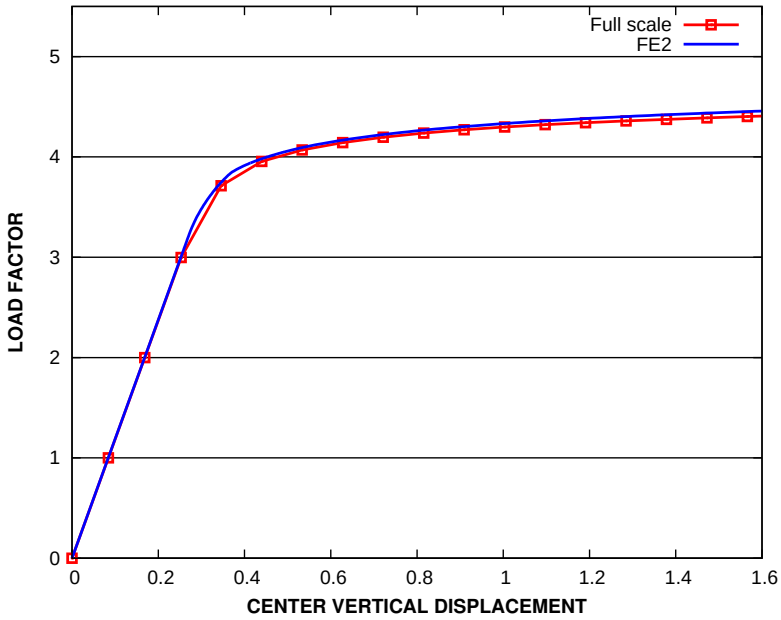


Figure 6.23: Load-displacement curve for full scale ($neq = 2.7$ million) and coupled models.

6.4 Hexagonal honeycomb sandwich cell

Sandwich panels with a core of hexagonally aligned cell walls, often called honeycomb cores in the literature, are extensively used in applications. However, the generation of meshes and identification of a periodic unit cell are somewhat more complicated than for the one- and two-directionally stiffened structures considered until now.

One of the additional difficulties when modeling hexagonal combs is that no obvious unit cell exists that allows to periodically attach instances of it to each other and generate a global mesh with smooth edges. For illustration, so-called a tripod unit cell is naively chosen as in Figure 6.24(a), with in-plane measurements l_x^T, l_y^T and height h. The superscript $(.)^T$ hints at quantities related to a tripod unit cell. After assembly of the global mesh (or a larger RVE depicting multiple closed honeycomb cells) in Figure 6.24(b), this choice of unit cell leads to overlapping edges instead of a rectangular boundary. In addition, every other column of cells has to be placed with a vertical offset of $l_y^T/2$ to correctly connect the respective cell walls. This is especially problematic when assembling multiple tripod cells into a larger RVE, where the number of cells is still small so that boundary effects strongly influence the results of the numerical homogenization and can not be neglected.

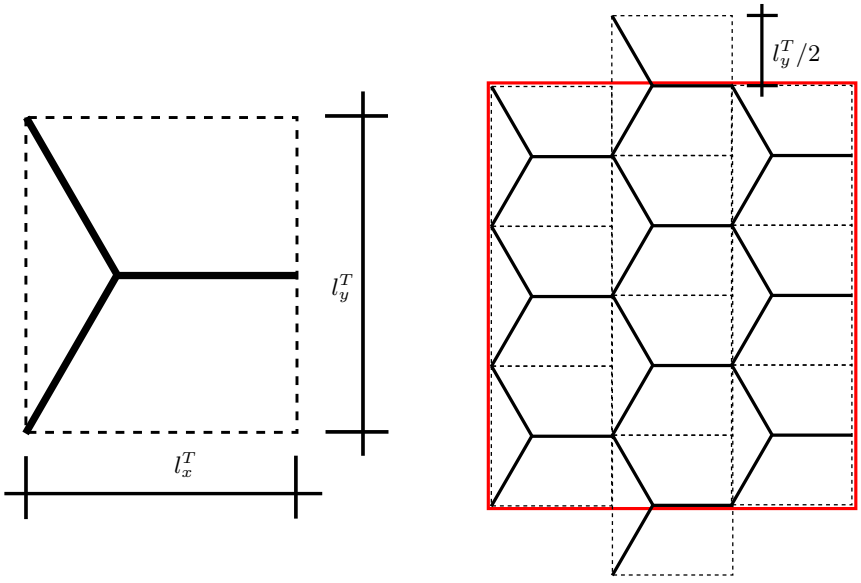


Figure 6.24: (a) Top-down view of a possible minimal "tripod" unit cell, leading to hexagonal cells when assembled.
(b) Assembled global structure (or larger RVE) with non-rectangular boundary.

In order to avoid overlapping edges and the need to offset certain columns of a grid-like assembly of unit cells, an RVE is chosen where each of the tripod's edges ends in one corner of the quadratic unit cell area. Without loss of generality, the top-right, bottom-left and bottom-right corners are chosen in a top-down view. This leads to a comb structure with irregular hexagonal cells, but allows to simply place cells next to each other as with the bidirectionally stiffened structure. When mapping the unit cell's bounding area in a top-down view to the square $[-1,1] \times [-1,1]$, the location of the point S at which the three cell wall segments intersect each other can be varied to generate different hexagon structures. It is noted that unsuitable choice of S could lead to degenerated cells such as rectangular cells (e.g. $S = (0,0)$) or non-convex hexagonal cells with interior angles greater than 180° (e.g. $S = (-1/3, 1/3)$). In the following, $S = (1/3, -1/3)$ is assumed as sketched in Figure 6.25(a). An assembled structure consisting of these tripod cells can be seen in Figure 6.25(b), which is now bounded by a rectangle without having to offset every second column of cells.

It is noted that the cell walls are no longer aligned along the in-plane x- or y-directions, but along three different in-plane directions in the $x-y$-plane instead. Thus, the previously used discretization parameters h_x and h_y have to be redefined to account for the new type of microstructure. Particularly, h_y is dropped and h_x now describes the number of elements in the lateral direction of any cell wall segment, i.e. the number of elements between any of the bottom-left, bottom-right, or top-right corner nodes and the point S.

When comparing the discretization of a minimal RVE in Figure 6.26, consisting of 56 elements, with the minimal RVE of the bidirectionally stiffened structure (Figure 6.8, 16 elements), it becomes clear that the number of equations in a classical shell model of a global structure is significantly higher when using hexagonal cells. This further reinforces the need for an improved model, but it also increases the difficulty of obtaining reference solutions via a classical simulation. In fact, the limits of FEAP are reached at about $3-4$ million equations. For this reason, some of the full scale reference solutions in this section are generated using the commercial finite element program Abaqus [1], allowing to run somewhat larger simulations.

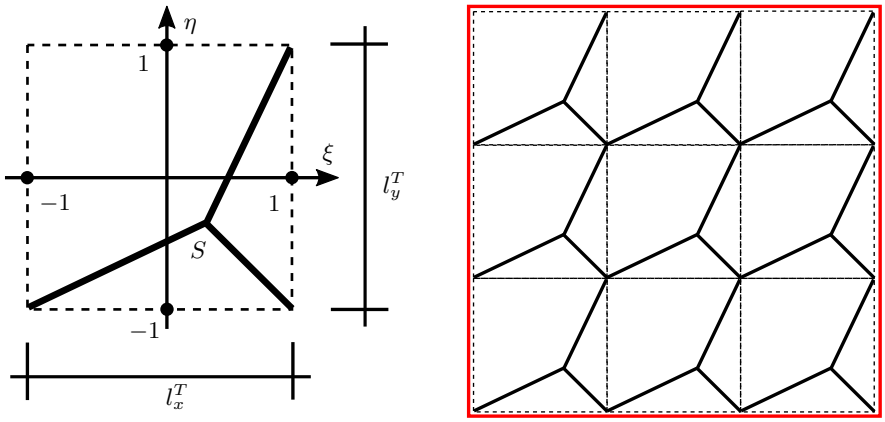


Figure 6.25: (a) Top-down view of a 1×1 tripod cell with $S = (1/3, -1/3)$, leading after assembly to hexagonal cells without need for offset or overlapping edges.
(b) Assembled global structure (or larger RVE) with 3×3 tripod cells and a rectangular boundary.

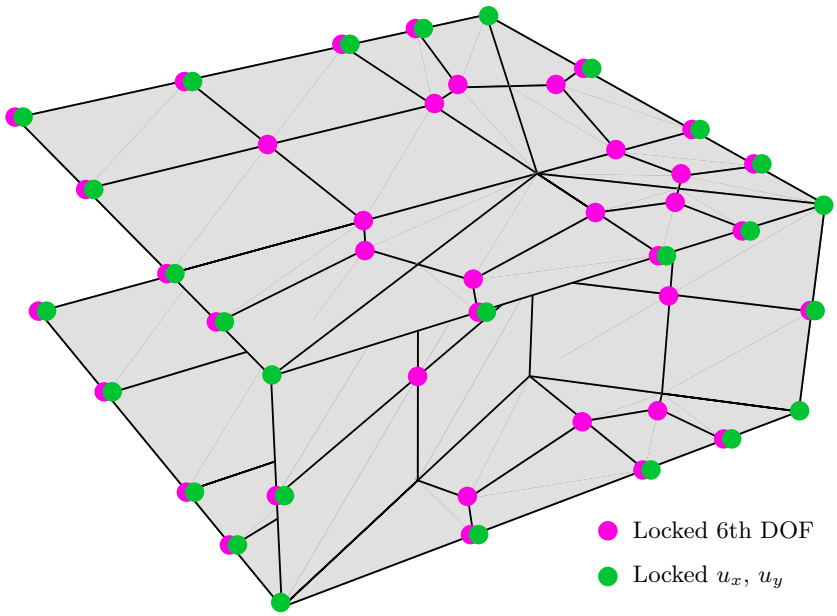


Figure 6.26: 1×1 tripod mesh with minimal discretization (lateral elements per cell wall $h_x = 2$, elements in thickness direction $h_z = 2$) and boundary conditions for the honeycomb structure.

6.4.1 RVE material matrix

As a first step for the new type of microstructure, the material matrix for an exemplary RVE is investigated. For this means, a microstructure is considered where each tripod unit cell has lengths $l_x^T = l_y^T = 2$ in x- and y-directions, a height $h^T = 0.8$ and shell thicknesses $t_{fl} = t_{cw} = 0.1$, while the linear material parameters are chosen as before ($E = 7000$, $\nu = 0.34$). It is noted that the actual RVE lengths l_x^R, l_y^R depend on the choice of the RVE, where multiple tripod unit cells can be placed next to each other.

For this type of RVE, the material matrix

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^m & \mathbf{D}^{mb} & \mathbf{D}^{ms} \\ & \mathbf{D}^b & \mathbf{D}^{bs} \\ \text{symm.} & & \mathbf{D}^s \end{bmatrix} \tag{6.16}$$

is now fully populated and contains further coupling terms in $\mathbf{D}^{ms}$ and $\mathbf{D}^{bs}$ between membrane, bending and shear modes. However, it is noted that these coupling terms are usually several orders of magnitude smaller than the entries in $\mathbf{D}^m, \mathbf{D}^b$ and $\mathbf{D}^s$. As an example, an RVE assembled from 2×2 tripods (i.e. $l_x^R = l_y^R = 4$) with a slightly increased discretization $h_x = 4$ is considered, for which the linearized stress resultants are listed in Table 6.4. The full RVE mesh is illustrated in Figure 6.27.

D_{i1}	D_{i2}	D_{i3}	D_{i4}	D_{i5}	D_{i6}	D_{i7}	D_{i8}
1745.0740	647.2961	57.4041	-5.18E-05	-1.92E-05	-1.69E-06	6.05E-07	-1.29E-06
	1745.0740	57.4041	-1.92E-05	-5.18E-05	-1.69E-06	1.29E-06	-6.05E-07
		597.8546	-1.59E-06	-1.59E-06	-1.75E-05	-7.24E-07	7.24E-07
			263.9696	91.8922	2.8943	-2.6405	-5.8557
				263.9696	2.8943	5.8557	2.6405
					89.1541	-2.5521	2.5521
	symm.					1462.6890	777.5331
							1462.6890

Table 6.4: Material matrix $\mathbf{D}$ for a 2×2 tripod RVE with $h_x = 4$, $h_z = 2$.

Some observations regarding this result are:

a) The diagonal entries D_{ii} are still dominant with $|D_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{8} |D_{ij}|$ for all i.

b) The membrane/bending and membrane/shear coupling terms in $\mathbf{D}^{mb}$ and $\mathbf{D}^{ms}$ are by at least 7 orders of magnitude smaller than the diagonal entries and can be neglected.

c) The bending/shear coupling terms in $\mathbf{D}^{bs}$ are still significantly smaller than terms in $\mathbf{D}^{b}$ (factor of at least 20) or in $\mathbf{D}^{s}$ (2 orders of magnitude).

When carrying out mesh refinement and investigating different discretization parameters h_x and h_z for the same 2×2 tripod RVE, it turns out that all relevant entries in $\mathbf{D}$ only vary by 1% or less, indicating sufficient accuracy already for coarse meshes.

A further investigation is carried out for the behavior of $\mathbf{D}$ for RVEs of different sizes, i.e. with a varying number of $n \times n$ tripod unit cells with $n = 1, 2, \ldots, 16$ (thus $l_x^R = l_y^R = 2, \ldots, 32$). Similarly to the uni- and bidirectionally stiffened structures considered in the last sections, the shear stiffness components in $\mathbf{D}^s$ can only be roughly approximated and vary for different RVE lengths. In order to show that the other relevant parameters remain constant, the matrix

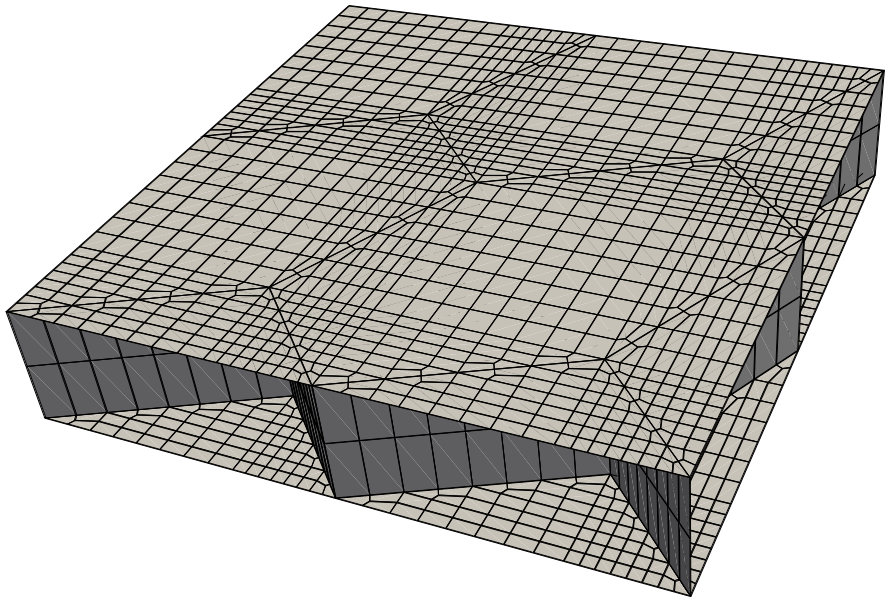


Figure 6.27: RVE mesh with 2×2 tripod unit cells and discretization parameters $h_x = 4$, $h_z = 2$.

$$\mathbf{D}' = \begin{bmatrix} \mathbf{D}^m & \mathbf{D}^{mb} & \mathbf{D}^{ms} \\ & \mathbf{D}^b & \mathbf{D}^{bs} \\ \text{symm.} & & 0 \end{bmatrix} \tag{6.17}$$

is introduced, representing $\mathbf{D}$ without the pure shear components in $\mathbf{D}^s$. Instead of investigating all 33 remaining matrix entries individually, including some irrelevant components which are very small, suitable matrix norms $||\mathbf{D}'||$ are introduced and compared for different RVE sizes.

For this means, the $L^{0.5}$, L^1 and L^2 matrix norms

$$\begin{aligned} ||\mathbf{D}'||_{L^{0.5}} &= \left(\sum_{i=1}^{8} \sum_{j=1}^{8} \sqrt{|D'_{ij}|} \right)^2 \quad , \\ ||\mathbf{D}'||_{L^1} &= \sum_{i=1}^{8} \sum_{j=1}^{8} |D'_{ij}| \quad , \\ ||\mathbf{D}'||_{L^2} &= \left(\sum_{i=1}^{8} \sum_{j=1}^{8} (D'_{ij})^2 \right)^{1/2} \end{aligned} \tag{6.18}$$

are investigated, in order to include contributions of all relevant components, while keeping the impact of small parameters moderate ($L^{0.5}$), low (L^1) or negligible (L^2), respectively. In Figure 6.28, the resulting norms of $\mathbf{D}'$ are plotted over the number n of tripod cells per side in an $n \times n$ tripod RVE.

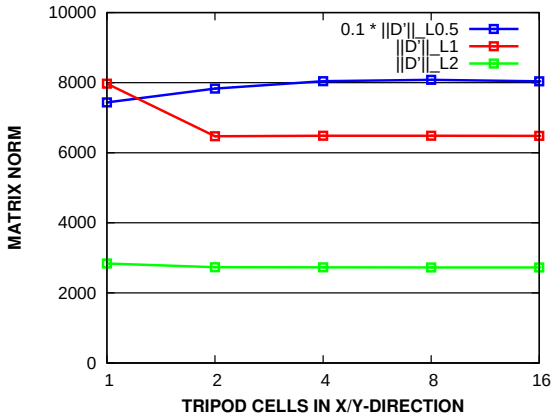


Figure 6.28: Matrix norms $||\mathbf{D}'||_{L^{0.5}}$, $||\mathbf{D}'||_{L^1}$ and $||\mathbf{D}'||_{L^2}$ for varying RVE sizes.

When comparing the different matrix norms, it can be seen that the L^2 norm is virtually constant, regardless of RVE choice, the L^1 norm converges fully starting from 2×2 tripod cells in the RVE, while the $L^{0.5}$ norm varies slightly as the non-dominant matrix entries have a much larger contribution to the norm than for L^1 or especially L^2. This indicates that the dominant stiffness components, especially on the diagonal of $\mathbf{D}'$, remain constant while only some smaller coupling terms differ for the various RVEs. This circumstance can also be verified when comparing some individual matrix entries at random for the different RVE choices.

6.4.2 Honeycomb sandwich plate

Having verified the convergence of linearized stress resultants for a honeycomb RVE in the last section, the next step is to model a full sandwich panel with hexagonal honeycomb cells and compare the results of the coupled simulation and a full scale reference solution. The structure that is modeled in this section closely relates on the global scale to the bidirectionally stiffened sandwich plate described in Section 6.3.2, but on the local scale the microstructure corresponds to a unit cell as shown in Figure 6.26.

Using standard units $[kN]$ and $[cm]$ as before, the measurements for the considered structure are chosen as $l_x = l_y = 40$, $h = 0.8$, with shell thicknesses $t = t_{fl} = t_{cw} = 0.1$ for the face layers as well as for the cell walls. Each tripod unit cell, as illustrated in Figure 6.25(a), has lengths $l_x^T = l_y^T = 2$ so that the full structure comprises 20×20 unit cells, or, due to open cells at the system boundaries, 19×19 closed honeycomb cells. Linear-elastic material data is chosen as before with $E = 7000$, $\nu = 0.34$, and the structure is loaded with a constant area load $q_z = 0.005$ at the top face layer (full scale simulation) or the global shell reference surface (coupled simulation). Unlike the bidirectionally stiffened system considered before, the honeycomb structure has no symmetries with respect to the x- or y-axes. Thus, the full system must be modeled instead of modeling only a quarter-plate. At the lateral boundaries $x = \pm l_x/2$, $y = \pm l_y/2$, all nodes have a fixed displacement $u_z = 0$, irrespective of their z-coordinate, representing a Navier-type soft support throughout the full thickness h. The full scale model with mesh, load and boundary conditions is illustrated in Figure 6.29, with the exception of inhibiting rigid body modes by locking u_x and/or u_y at two additional nodes.

Similar to the bidirectionally stiffened plate, the analytical result for a homogeneous plate $w_{\max} = 0.004055\, q_z l^4/K$ can be used to generate a very rough first approximation, using the bending stiffness parameters $D_{44} = D_{55}$ from Table 6.4 for the plate stiffness K. Inserting the parameters $q_z = 0.005$ and $l = 40$ from this example leads to the

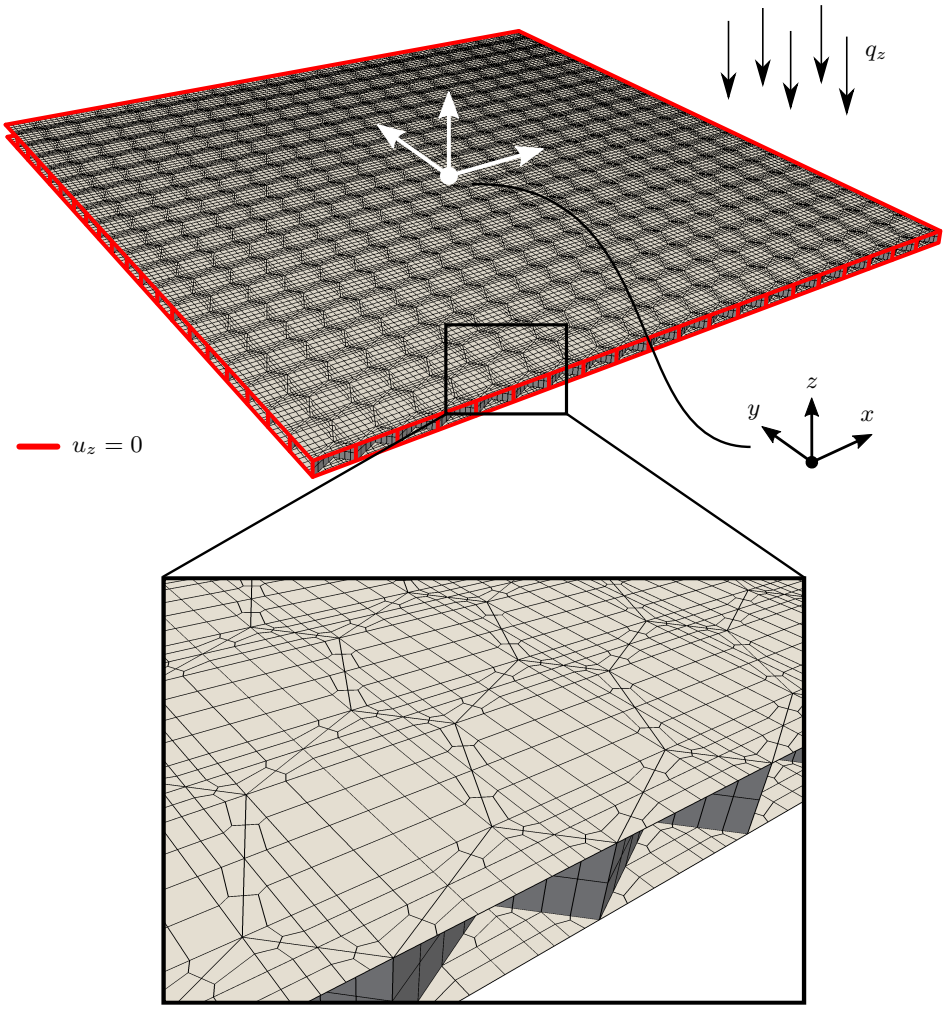


Figure 6.29: Full scale model (no symmetry) for the honeycomb sandwich plate.
The system comprises 20×20 tripod unit cells with discretization parameters $h_x = 4$ (elements per cell wall segment in length direction), $h_z = 2$ (elements through thickness). On the outer boundaries, $u_z = 0$ is imposed as a boundary condition, with additional conditions at single nodes to prevent rigid body modes (not drawn). The structure is loaded with a constant area load q_z on its upper face layer.

approximation for the homogeneous plate $w_{\max,\hom} = 0.19663$.

Convergence of the results is established by comparing full scale and coupled models with varying discretizations. For the full scale reference models, discretization parameters h_x and h_z are increased to achieve finer meshes. Three different systems with $h_x = 2 \cdot h_z = 2, 4, 8$ have been investigated with FEAP, while the solution for $h_x = 12, h_z = 8$ comprised 3.5 million equations and was computed using Abaqus using the element S4R. For the coupled simulations, mesh refinement was carried out by increasing the number of $n \times n$ elements on the global scale, within whose integration points a fixed RVE with 2×2 tripod unit cells and discretization parameters $h_x^R = 2$, $h_z^R = 4$ was evaluated.

The deformed mesh for one of the full scale simulations is plotted in Figure 6.30. All calculation results are summarized and compared in Table 6.5, along with the respective discretization parameters and the total number of equations neq. It is noted that for the coupled model $neq = neq_{glob} + 1 \cdot neq_{loc}$ holds, since for the linear case it is sufficient to evaluate the local model only once instead of once per integration point. This is evidently not the case for nonlinear models, according to previous examples.

Both the coupled and the reference models lead to an agreeing maximum displacement, see Table 6.5, as well as equivalent deformation figures. Again, the simulation results with $w_{\max} = 0.2222$ exceed the analytical approximation for the homogeneous plate $w_{\max,\hom}$ by a slight amount (roughly 13%). In the full scale models mesh refinement leads to further increases in maximum displacement, even when beginning with a fine discretization. It was not possible to generate a fully converged full scale solution due to memory limitations. Using the coupled model, the required number of equations can be drastically reduced, which leads to fast calculation times and low memory requirements. It is noted that the finely discretized full scale models already imply considerable hardware requirements just for the graphical output of the system, both when modeling as well as in post-processing.

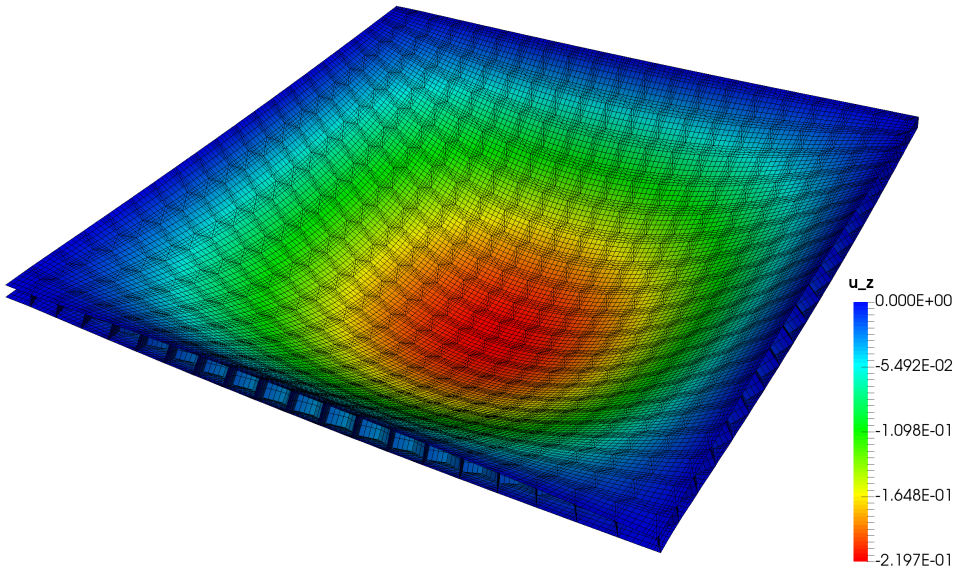


Figure 6.30: Deformed mesh (50× amplified) and vertical displacement (contour) for the full scale model with $h_x = 4$, $h_z = 2$.

a. Coupled model

$n \times n$	neq	$w_{max}\,[cm]$	$t_{solv}\,[s]$
4 × 4	3499	2.076E-01	< 0.1
8 × 8	3763	2.145E-01	< 0.1
16 × 16	4771	2.193E-01	< 0.1
32 × 32	8707	2.214E-01	≈ 0.1
64 × 64	24259	2.220E-01	0.44
128 × 128	86083	2.222E-01	1.89
256 × 256	332611	2.222E-01	9.78

b. Full scale model

Program	h_x	h_z	neq	$w_{max}\,[cm]$	$t_{solv}\,[s]$
FEAP	2	1	93046	2.169E-01	2.95
FEAP	4	2	337727	2.197E-01	18.41
FEAP	8	4	1283089	2.203E-01	123.46
Abaqus	12	8	3538812	2.207E-01	956

Table 6.5: Maximum vertical displacement and solution time for different coupled (a) and full scale simulations (b).

Chapter 7

Mesh generation

The coupled global-local algorithm proposed in Chapter 5 was implemented by means of a constitutive model for the FEM program FEAP, which is well-suited for this approach as the whole source code was available without having to cope with possibly limiting interfaces provided by commercial finite element programs. FEAP is relatively popular in the research environment for the simulation of academic examples, which most often do not require to model complicated meshes. However, this is not given in this work, where the microstructure of sandwich shells must be modeled both in terms of an RVE for the coupled method, as well as for generating classical shell models as full scale reference solutions. While some basic functionality for generating blocks of elements with arbitrary discretization granularity is provided by FEAP, there is no way to automatically generate multiple blocks in a regular pattern, such as blocks of individual cell walls or cell wall segments in a comb-like sandwich core.

In addition to the generation of a finite element mesh with nodes and element connections, special attention must be devoted to the 6th degree of freedom, which is only present at nodes with shell intersections and must be locked otherwise, see (4.18). As FEAP provides no means to automatically set boundary conditions on intersection or non-intersection nodes, these nodes must be manually determined and endowed with the correct boundary conditions.

For these reasons, it is inevitable to find a way to generate shell discretizations as used in the examples in Chapter 6 in an automated fashion, matching sandwich core and face layers to each other and setting correct boundary conditions for the 6th degree of freedom. Two different methods have been used for the examples in this work and will briefly be presented in the following.

7.1 Automated mesh generation for FEAP using Python scripts

The most sophisticated routine the FEAP input file syntax provides for the automated generation of groups of elements is the `bloc` command. With its help, it is possible to generate a single structured block of elements with flexible discretization relatively easily. However, a block of shell elements must lead to a smooth and coherent surface, so that it is not possible to generate cell core structures for any of the examples in Chapter 6 as a single block. Already in the simplest example, the axially stiffened sandwich, multiple disjoint cell walls must be created, requiring an individual `bloc` command for each of them as no loops can be used inside of a FEAP input file.

While in theory the `bloc` commands could be copied and pasted for each cell wall or honeycomb tripod leg, this method would be unsuitable in practice, where for every block all input parameters, such as position coordinates and element/node numbers, must be adjusted, which is arduous and prone to error. As a remedy, several mesh generation scripts were implemented using Python [79] in order to automatically generate ready-to-use FEAP input files both for the RVEs as well as the full scale reference models covered in Chapter 6. These scripts are used for the first two groups of examples, the uni- and bidirectionally stiffened sandwich structures, while for the hexagonal honeycomb sandwich a different method will be described further below.

Inside the program, the user only needs to provide some parameters for geometry, discretization, material and load case (see Table 6.1 for an overview). The script will then generate an input file for FEAP, containing header (`feap` specifications and choosing a solver), footer (material definitions, `macr` commands for a full scale model or `batc` commands for an RVE) and – most importantly – automatically generated `bloc` commands for each of the n_y (axially stiffened sandwich) or $n_x \times n_y$ (bidirectionally stiffened sandwich) cell walls. Moreover, one `bloc` for each face layer is created with a matching discretization, meaning that nodes in the face layers are present at locations where the cell walls intersect the face layers and thus the parts can be connected by tying the intersection nodes. Finally, all nodes with cell wall/cell wall or cell wall/face layer intersections are identified and endowed with the correct boundary conditions for the 6th DOF.

This allows to generate complete full scale or RVE models with arbitrary spatial dimensions and any number of stiffeners in a matter of seconds, without dangers of incorrect parameters or errors when manually copy-/pasting from similar input files. For instance, all that was necessary in order to modify the bidirectionally stiffened plate example in Section 6.3.2 (16×16 stiffeners with thickness 0.1 $[cm]$ placed in regular intervals of 2 $[cm]$) towards the much finer microstructure (Section 6.3.5.c,large scale model with 128×128

stiffeners, thickness $0.025\,[cm]$, placed in intervals of $0.25\,[cm]$), was to change the parameters for n_x, n_y, l_x^R, l_y^R and t_{cw} in the mesh generation script and re-run the program to generate the new input file.

The program code introduces several general purpose classes for entities such as elements, nodes, `blocs`, materials and more, which can easily be used or modified to be able to generate arbitrary meshes in an automated way.

7.2 Conversion of Abaqus meshes using inp2feap

For the hexagonal honeycomb meshes investigated in Section 6.4, a different method was employed due to multiple additional challenges related to the mesh generation:

- The full scale reference solutions for the hexagonal cell meshes lead to a significantly larger equation system in order to be able to calculate a converged solution. Using FEAP, it was only possible to simulate models with up to 2 - 3 million equations, a limit easily excessed by the full scale models of these structures. Consequently, a conforming mesh had to be generated for an alternative FEM program, for which Abaqus was chosen.

- In order to fit the face layer meshes to the cell walls, irregular (non-rectangular) elements in the face layers are required - see Figure 6.26, for instance. Additionally, any triangular shaped elements must be decomposed into three quadrilaterals, in order to preserve compatibility with the employed shell element. Thus, the face layers can no longer be generated by simple block-statements in FEAP but must instead be automatically generated by a more sophisticated script.

- Similarly, the hexagonal cell walls can no longer be generated using a single block per cell wall, spanning from $x_{\min}$ to $x_{\max}$ (or $y_{\min}$ to $y_{\max}$, respectively), but must instead be generated individually for each tripod unit cell with three blocks.

- Locking the 6th DOF on all intersection nodes would require multiple `edge` commands in FEAP for each honeycomb cell, quickly amassing to thousands of such commands in a full system. Each `edge` statement represents a search for nodes within a spatial cylinder in the set of all nodes. The time for mesh initialization using the method described in the last section already took as long as half an hour in some cases for the bidirectionally stiffened sandwich. In the case of honeycomb meshes, this issue is further intensified as both the number of nodes as well as the number of necessary `edge` commands are significantly higher for a comparable

structure.

Due to these reasons, an alternative method had to be found to generate meshes for the hexagonal honeycomb cell structure. Consequently, the mesh generation process for these systems has been broken down to a multi-step workflow involving several different tools:

1. Run a Matlab script to explicitly generate a minimal 1×1 tripod unit cell with customizable discretization fineness as an input file for Abaqus.

2. Use Abaqus CAE to assemble a complete mesh, either for an RVE (which might consist of a single or multiple tripod cells) or for a full scale solution where the model comprises many unit cells.

3. Run a Python script, `inp2feap`, to convert the fully assembled mesh from Abaqus into an input file for FEAP, or, alternatively, use the assembled model, additionally endowed with loads, boundary conditions and material data, for a large scale reference solution within Abaqus.

For the first step, a Matlab script was implemented where for given length parameters $l_x = l_y$ and h as well as discretization parameters h_x (number of elements along the length of one tripod "leg" segment) and h_z (number of elements in thickness direction) an Abaqus input file with explicit nodal coordinates and element nodal connections is generated. As an example, the RVE in Figure 6.26 was generated with parameters $l_x = l_y = 2$, $h = 0.8$ and $h_x = h_z = 2$. Additionally, all nodes where shell intersections are present or will be present when attaching further tripod cells to the mesh are identified and collected in a node set, to be able to cope with the boundary conditions necessary for the 5/6-DOF shell formulation introduced before.

This input file is further processed within Abaqus CAE in the second step, which is used to merge nodes which are overlapping in one spatial point and thus connect the five formerly separated parts (upper and lower face layer plus three cell wall segments). When generating a mesh for a full scale model or an RVE with more than 1×1 tripod cell, Abaqus is then used to assemble multiple instances of the tripod unit cell in a grid-like pattern, see for instance Figure 6.27 for a 2×2 RVE grid. All nodes mutual to multiple unit cells are merged, so that a connected mesh is created.

When generating a model for a full scale reference solution within Abaqus, the mesh is endowed with material parameters, boundary conditions and loads as a next step, followed by a linear or nonlinear solution process, respectively. Alternatively, the mesh can be exported to FEAP using the custom-built program inp2feap, which is especially required for all RVEs that are used in a coupled simulation. But also some full scale

reference solutions with moderate discretization fineness can still be generated by FEAP, as long as the system size does not exceed $2-3$ million equations. For this means, the model generated within Abaqus CAE is written out as a job file. It contains node and element data, as well as a node set for intersection nodes and an element set for each of the three constituents (upper face layer, lower face layer, and cell walls), so that different material numbers can be assigned to each part within FEAP.

The functionality of inp2feap for these specific honeycomb models can be summarized by the following main steps:

- Parse nodes (id, coordinates) and elements (id, nodal connections) from the Abaqus job file and convert them to FEAP syntax.
- Parse the node set for intersection nodes and generate a FEAP `boun` command for each contained node that explicitly unlocks the 6th DOF on that node. On all other nodes the 6th DOF is locked. `boun` leads to a considerably faster mesh generation when compared to `edge`, as the nodes are specified directly by their node id.
- Parse element sets and assign a specific material number to each element contained in a set. This allows to utilize different materials for the face layers and the cell walls, for instance.
- Write a FEAP input file with a specified header, the main part (comprising node and element definitions plus 6th DOF boundary conditions), and a footer (comprising material definitions, loads and further boundary conditions).

The exact behavior of inp2feap is controlled by input scripts following the JSON-syntax, which are documented along with the rest of the functionality on the project's Github page [30]. Here, the source code and some example files are available as well. Potentially any Abaqus mesh can be converted to FEAP, provided that a compatible element exists, such as the four-node quadrilateral in this case. It is noted, however, that inp2feap currently only parses nodes, elements, node sets and element sets. Any other model data, such as boundary conditions, loads, and materials, must be provided explicitly (e.g. in an appropriate header or footer), or the functionality of inp2feap must be extended.

Chapter 8

Parallelization and effective data transfer in a finite element program

Applying the solution methods for linearized boundary value problems which are presented in this work to real-world problems leads to linear equation systems with a large number of unknown variables. For instance, the full scale discretizations used as reference results can quickly yield linear systems with millions of unknowns. In order to be able to numerically solve these large systems, the usage of optimized codes is necessary, which strive to utilize the available hardware resources to maximum efficiency.

When developing such codes, it is inevitable to investigate the detailed functionality of the hardware, whereof some basics will be described in this chapter. With regards to currently available machines, it is especially expedient to use parallel programming, without which only a fraction of a processing unit's resources can be utilized. However, the development of effective parallel codes is subject to many additional difficulties when compared with sequential programs, even simple algorithms are in many cases not trivial to implement.

The parallel implementation used in this work has been originally proposed and was developed in [83]. Building upon that, the code has been improved in terms of stability by locating and eliminating several possibly occurring race conditions and deadlocks, parallel computing pitfalls that will be briefly described below. In addition, a shared memory implementation for inter-process communication was developed, leading to significant performance gains during data transfer in coupled global-local models. As a prospect, further exploitation of shared memory resources to quicken multi-scale finite element calculations will be possible in several ways, some of which will be discussed.

8.1 Parallel computer architectures

As the most simple starting point to describe computer architectures, the von Neumann model is often employed. It introduces a computer consisting of three main components: Memory, a control unit (CU) and an arithmetic logic unit (ALU). All of these parts are connected to each other. Instructions and data are not distinguished in the von Neumann model, both are stored in the same memory. In operation, the control unit sequentially fetches one instruction at a time from memory, decodes and executes it, possibly using the ALU to process data from memory. This is known as the FDX cycle (fetch-decode-execute). In this model, data is used to produce output according to rules specified by the instructions, so that for a given set of instructions and data, the output is always the same. The von Neumann computer is consequently said to be deterministic.

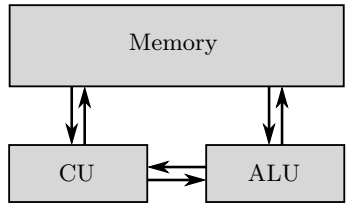


Figure 8.1: The von Neumann architecture scheme

For the classification of parallel computer architectures, Flynn's taxonomy [19] is commonly used, where micro architectures are distinguished depending on the amount of available instruction and data streams in a processing unit. Using the tokens S (single), M (multiple), I (instruction) and D (data), the architecture classes SISD, SIMD, MISD and MIMD are obtained.

Consequently, SISD architectures (single instruction, single data) are machines that, at one point in time, can only process a single instruction, working on a single data set. This is the simplest architecture type, corresponding to the von Neumann model, and has historically dominated the development of hardware for a long time, where systems with a single processor, comprising a single core, were generally prevalent. Intel, the world's leading CPU manufacturer, had introduced its first multi-core processor Pentium D not until 2005 [45].

SIMD machines (single instruction, multiple data) only execute a single instruction at a given time as well, but this instruction is allowed to process multiple data operands in parallel. Therefore, SIMD designs are especially well-suited for highly parallel, computationally intensive algorithms with large amounts of data, where relatively little program

logic is required and the control flow is simple. Practical applications are for instance large dense matrix operations which are known to occur in image manipulation or signal processing. In this regard, it is evident that modern graphics cards are a good example for SIMD architectures. Their technical development has been strongly driven by the video gaming industry, where millions of pixels have to be calculated and displayed in fractions of a second. But modern CPUs do provide SIMD instructions as well, by the means of vector-valued operations in special registers with extended widths of 128 – 512 bits, where for instance several floating point operations can be executed with a single instruction. These extensions of the instruction set architecture are known as SSE (Streaming SIMD Extensions) or AVX (Advanced Vector Extensions), for example.

The MISD architecture (multiple instructions, single data) is relatively uncommon, it allows in theory to execute several instructions at once, each of them operating on the same single data set. Applications for this architecture can be found in redundant calculations sometimes used in error detection and correction, but it is of little use for most other practical or scientific purposes.

In modern computers, the MIMD architecture (multiple instructions, multiple data) is most widespread, it provides for both multiple instruction streams and multiple data streams, so that different operations can be processed on different data at the same time. Thus, MIMD is the most flexible architecture. Nearly all modern microprocessors for PCs, workstations and even cellphones can be related to it. In the perspective of high performance computing, MIMD machines have been available since a longer time, where in so-called supercomputers many single machines (nodes) are connected via high-speed interfaces to clusters, which can contain a total of many thousands up to millions of processing cores.

With this background in mind, it is evident that the development of parallel codes is an important aspect of scientific computation. Some basic concepts of parallel programming and computation will be presented in the following subsections.

8.1.1 Increase in computational power due to more processing cores

Technological advances in hardware design have manifested themselves in the form of rising CPU clock rates for several decades. However, a paradigm change can be observed since a few years, where multi-core processors have become increasingly widespread and comprise more and more cores per chip, while clock rates tend to stagnate somewhere between 2 and 4 GHz. Current high end server CPUs such as the Intel Xeon E7-8890 v3 (Haswell-EX) contain up to 18 physical cores on one chip. When including Simultaneous

MultiThreading (SMT) / Hyper Threading Technology (HTT), the operating system has up to 36 virtual cores at its disposal for a single chip, or up to 144 virtual cores in a machine with the maximum number of 4 chips.

The main reason for turning away from ever increasing clock rates is the increase in power consumption of a processor in that relation, which leads to high amounts of generated heat that can damage or destroy the hardware. Consequently, the thermal design power (TDP) of a processor has become one of the most important design criteria for new hardware generations, not only due to ecological reasons. In the present, a trend can be observed where between two hardware generations the computational potency of a chip increases, while its power consumption decreases at the same time.

But even with efficient chips, power leakage poses an increasingly difficult challenge to cope with for hardware design, where transistor sizes progressively decrease with time and the amount of transistors per unit area increases, as stated by Moore's law [69]. For some time past, it is assumed that the advancement in micro chip design is no longer limited by manufacturing techniques – Intel predictions in correlation with Moore's law reach out to the year 2020, where transistors with a feature length of 5 nm are envisioned – but by the amount of dissipated thermal power instead [8, 77]. As a result, it becomes increasingly important to develop efficient parallel codes in order to enable full utilization of hardware resources.

8.1.2 Memory-bound applications

Another important aspect for developing effective codes lies within the fact that the execution speed of most scientific algorithms is not limited by the amount of necessary integer or floating point arithmetic operations, but instead by latency and bandwidth of the system memory, which determine the amount of time in which the data required for arithmetic operations can be provided. In fact, even individual processor cores can potentially execute many floating point operations in a single clock cycle by utilizing mechanisms such as instruction level parallelism (ILP) or vector-valued SIMD instructions. On the other hand, a single query to data in system memory usually requires up to several hundred CPU cycles. Subsequently, memory access times are the bottleneck for the execution speed of many codes with the actual arithmetic operations being comparably insignificant.

For this reason, modern chips tend to invest increasingly large parts of their transistor budget (see Moore's law) to provide for fast SRAM cache memory in different hierarchy levels with as much capacity as possible. Utilizing these to full benefit when implementing

an algorithm, being aware of cache line functionality and using techniques such as cache-blocking, can potentially greatly reduce the amount of required main memory accesses and thus increase the execution speed in memory-bound programs.

8.1.3 Memory topologies in parallel machines

Particular attention has to be devoted to the memory topology in parallel computer architectures. Machines in which all processing cores share equal access to the same memory are referred to as shared memory processing (SMP) machines. In this category particularly nearly all desktop machines can be found, where even in multi-core systems each CPU core has equal access to the shared system memory. The alternative model is given by distributed memory machines, where, for instance in a cluster, each node has its own main memory and direct access to the memory of other nodes is not possible. While communication between nodes is made available by network or interconnect interfaces, these means of transfer feature significantly worse latency and bandwidth than the direct access to system memory resources. For this reason, when developing codes for distributed memory systems and clusters, it is essential to consider and exploit the topology of nodes and memory hierarchies in order to minimize the occurrence of slow data transfers.

The Message Passing Interface MPI [62] provides a standard which allows for inter-process communication on a single node or on different nodes of a cluster, while OpenMP [73] provides for the simultaneous utilization of all processing cores on shared memory systems. In more current versions of the MPI standard, starting with version 3.0, explicit shared memory support is introduced as well, so that MPI is in theory the more flexible tool. However, implementing parallel algorithms for shared memory machines, especially those based upon already available sequential code, proves to be much easier using OpenMP, which is why it is also used in the code underlying this thesis.

8.1.4 Potential benefits and drawbacks of hardware accelerators

The term GPGPU (General Purpose computation on Graphics Processing Unit) describes the offload of certain computational work, for example a scientific algorithm, from the CPU to a graphics processing unit (GPU). As a more general term, GPUs and closely related non-CPU computation devices such as the Intel Xeon Phi coprocessors [48] are also known as hardware accelerators. These devices allow for massively parallel high performance computations by means of the SIMD architecture, they can significantly beat

CPUs in terms of theoretical computational power, measured in floating point operations per second (Flops), and in terms of the memory interface bandwidth as well. However, algorithms with nested control flow and low arithmetic density, meaning few arithmetic operations per data set retrieved from memory, can prove to perform badly using GPU computing. On a GPU, scalar multiprocessors (SMs) are the rough equivalent of a CPU core. While each SM features many arithmetic-logical units in comparison to a CPU, it has a low amount of control units and can not efficiently process complex program flows with different operations for different data in the sense of MIMD.

Another distinctiveness lies within a GPU's dedicated memory, which is separated from the system memory and usually turns out being somewhat smaller. For instance, the high end scientific computing card NVIDIA Tesla K40, which was introduced in November 2013 [72], comprises 12 GB of video memory (device memory), whereas a CPU-based system at this time can easily feature 128 GB of memory or more. The GDDR5 video memory is interfaced to the SMs with a high bandwidth of 288 GB/s (Intel Ivy Bridge: approximately 100 GB/s), but data must be transferred from the system memory to the device memory via the PCI Express bus before it can be processed. This is very slow in comparison, featuring a maximum bandwidth of 16 GB/s as of the current standard PCIe 3.0 x16.

For this reason, data needed for computation on a GPU should be kept and further processed on the device memory as long as possible, often requiring the re-implementation of large parts of the program structure to be operating only on the GPU. This can not only prove extremely time-consuming when modifying complex programs, it can also lead to performance issues in program parts with low arithmetic density.

Despite the theoretically superior computational power of a GPU, its utilization is thus no suitable alternative for the code used in this work.

8.2 Parallel programming issues and pitfalls

8.2.1 Race conditions

Due to the fact that in parallel programs the same code is executed at the same time by several threads, data access conflicts can possibly arise when not paying attention on the exact access patterns to shared data. The result of an algorithm can in this case depend on the order in which the different threads execute critical instructions. This circumstance is referred to by the terms race condition or data race. For example, it is

a possible situation that two threads A and B must load the same variable, increment it by one each and write it back to memory. However, if thread B reads the variable from memory before thread A was able to write back its result, a wrong value is read by thread B, incremented and then written back to memory, so that one of the increment operations is lost. This situation is known as a write-write conflict. Similar types of conflicts can be found in read-write conflicts, where wrong data is read and processed by a thread.

In order to avoid such conflicts when accessing shared data, the access patterns of multiple threads must be synchronized, for which tools are provided by OpenMP in the form of `barrier`, `critical` or `atomic` statements, for instance. It is important to keep in mind that these synchronization operations cause different threads in a program to wait for each other at specific locations. The code is effectively serialized in parts. However, the performance loss in this context has to be condoned to preserve calculation of correct results.

In practice, it is nearly impossible to find all potential race conditions in a complex program by manual source code inspection. Additionally, they can lead to wrong calculation results instead of freezing or crashing programs, which is presumably an even worse situation. There are tools built on purpose to find possible conflicts such as the Intel Thread Inspector [47]. Using automated analyses provided by such tools is essential to be able to develop correct parallel codes.

8.2.2 Deadlocks

Using synchronization statements and barriers introduces a new type of potential issue in parallel programs which is called deadlock. In this situation, one or multiple threads or processes are waiting for the release of a locked resource or at a barrier, which can only be resolved by another thread or process that is currently waiting itself. Thus, the lock is never resolved and the program freezes while multiple threads/processes are waiting for each other. This issue can easily emerge when not carefully implementing synchronization commands, which are on the other hand required to prevent the occurrence of data races.

Unlike many simpler sequential programming mistakes and analogously to data races, potential deadlocks do not occur independently of the used data and can be extremely hard to pinpoint by manual investigation. Again, purpose-built parallel debuggers and programming tools can and must be used in order to find and resolve deadlocks.

8.2.3 Determinism of floating point arithmetics

In a computer, real numbers and all constructs made up of them, such as vectors and matrices, are expressed in a floating point representation. This representation only has finite precision, depending on how much memory is used to describe a single number – 4 bytes for single and 8 bytes for double precision numbers are commonly used. As many numerical algorithms are susceptible to rounding errors, calculation using double precision numbers is widely spread in scientific applications, despite higher memory requirements and computation times when compared to single precision.

Arithmetical operations defined on floating points numbers imply some specific issues already in sequential programs. Depending on the condition of a mathematical system, rounding errors can occur and do in some cases pose a large problem. In parallel programs, an additional issue arises as floating point additions and multiplications are not associative, it is $(a + b) + c \neq a + (b + c)$ in the general case. Since the execution order of instructions in parallel code is often not determined (see 8.2.1), the results of some parallel algorithms are not deterministic, they can produce different output values for the same input. As an illustrative example, the parallel calculation of a residual vector norm can be investigated. Here, several threads sum up individual vector components in an arbitrary order, leading to slightly different results when the execution order changes. Contrary to the issue of race conditions, this fact is normally unproblematic. The numerical summation of several floating point numbers is in any case only an approximate solution to an analytically exact summation of real numbers. While the result depends on the order of execution, there is no right or wrong order per se, neither in the parallel, nor in the sequential case. This issue is just not visible in most sequential programs, where the order of execution is fixed.

Using special implementations such as the addition of partial sums, the omnipresent problem of rounding errors can be reduced in some cases.

8.2.4 Address locality and inter-process communication

In parallel computing, two different types of execution streams are distinguished. A single program can possibly have multiple threads which all use the same address space and access the same memory, whereas processes can be regarded as individual programs on their own, managing their specific private virtual address space. Processes are usually unable to access data from other processes directly. Implementing algorithms under the utilization of multiple processes is especially common when using clusters with the help

of MPI. In this case, the separated address spaces reflect the topology of the underlying hardware, where different nodes have no access to any shared memory and can only communicate by message passing via the network. But even when using a shared memory machine, implementing a program using multiple processes can be beneficial due to ease of implementation and potential portability to distributed memory architectures. This concept is also employed for the coupled method in this work.

When implementing parallel programs using multiple processes, special attention has to be devoted to the communication between processes. Within the scope of this thesis, an implementation has been developed which allows for direct data transfer via the system memory. On shared memory machines, this leads to a significant performance increase for data transfer when compared to previous implementations, where pipes or even slow hard disk drive access was utilized. The proposed implementation is provided as an optional module and will be referred to as the shared memory extension of FEAP. It will be discussed in more detail in the next section.

8.3 FEAP shared memory extension

The algorithm introduced in this work leads to one local boundary value problem for every integration point of the global shell model, all of which are independent from each other. Thus, they can be solved in parallel. However, the local systems require the local strains of the global system to be provided, while in turn the global system must receive the local stress resultants and material tangents from the local systems in each new iteration or load step. In addition, in cases of local physical nonlinearity such as von Mises plasticity, history data must be loaded and stored for each integration point in each time step, which can lead to substantial amounts of data. The largest simulations that are carried out for this work generate up to 15 GB of history data, which must be read and rewritten once in each iteration step. For this reason, it is expedient to investigate possibilities for an effective data transfer in global-local scale communication and quick access to saved fields of history variables.

In the present implementation with FEAP the global and local systems will be solved by different processes. The number of parallel processes on the micro scale usually complies to the number of available processor cores, additionally capitalizing on hyper threading virtual cores as introduced in Section 8.1.1. It has been explained in the last section that each process operates in its own distinct address space and that inter-process communication requires an explicit transfer of data, preferably using the fastest option available to the hardware. As the interactive version of FEAP that is used in this work is limited in

scope to only support shared memory machines, utilization of the system memory constitutes the fastest available option for data transfer. In order to provide an implementation featuring inter-process communication by means of the system memory, the Windows API was employed to allow processes to open memory mappings which can be read from or even written to by other processes, using routines such as `CreateFileMapping` [64] and `MapViewOfFile` [65]. These mappings point directly in the system memory and allow full utilization of its relatively low latency and high bandwidth, when compared to prior alternatives such as hard disk transfer.

On the downside, the implementation in this form is platform dependent and requires the used compiler to provide functioning access to platform routines, which was not given using the Silverfrost FTN95 compiler [87] in the tested version. For this reason, the shared memory extension can be optionally deactivated to preserve compatibility. A possible alternative method of implementation would be to use a more current MPI version which allows for explicit access to shared memory interfaces. However, at the time of implementation of the shared memory scheme, the MPI 3.0 standard, which introduced shared memory functionality, had not been published yet. Additionally, the execution of MPI programs requires to use a specific MPI runtime environment `mpirun` and is thus more complicated to use, which is not the case for the Windows-specific implementation.

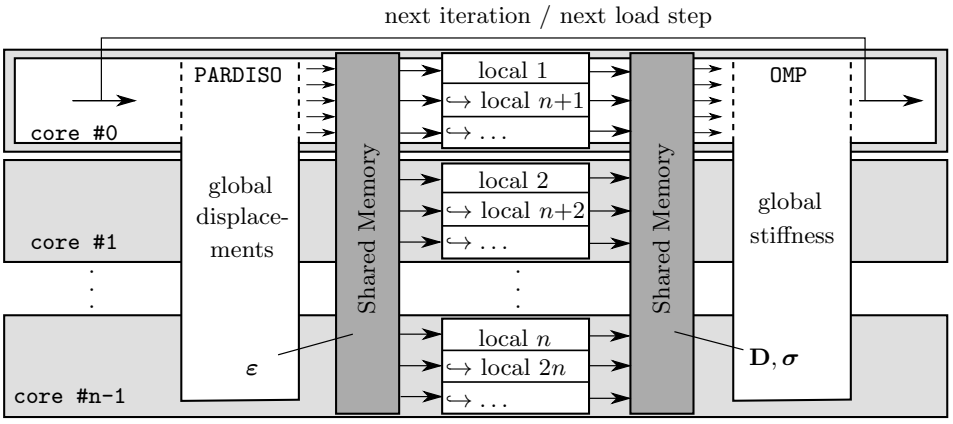


Figure 8.2: Shared memory program scheme for global and local processes.

A rough draft of the shared memory multi-scale implementation is shown in Figure 8.2. In the vertical direction, each light gray box depicts a single processor core provided by the hardware. Typically 8, 12 or 16 cores are used in current machines, even though much higher numbers would theoretically be possible in a single workstation, see 8.1.1. It is assumed in this figure that n cores are available. Each white box describes a single

process, thus, core `#0` hosts both the process for the global scale system, which will also be referred to as the host process, as well as local processes 1, $n+1$, $2n+1$, ..., related to one specific integration point of the global system at a single time. In some parallel regions the host process is extended to all available cores with the help of multithreading, e.g. for solving the equation system in parallel using the PARDISO solver [57, 81, 82], or for assembling the global tangential stiffness matrix. For the local scale problems, cores `#1` to `#n-1` host a single process each at any given time, evaluating one specific local boundary value problem. In order to provide necessary data for the local scale problems, the host process on core `#0` creates a mapping into the system memory where the local shell strains $\boldsymbol{\varepsilon}$, derived from the global displacements $\mathbf{u}$, are stored. Additionally, memory is allocated to which the results of the local calculations will be written later.

Afterwards, a series of n processes, depicted by white boxes, is created, each of them correlating to a single integration point of the global system. Each local process i obtains its specific local shell strains $\boldsymbol{\varepsilon}$ from the previously created memory mapping, calculates the local material tangent $\mathbf{D}$ and stress resultants $\boldsymbol{\sigma}$ and stores them in the memory location provided by the host process. The process i is closed afterwards and a new process, corresponding to a new global integration point $n+i$, is created, until all integration points have been taken into account.

At this point, only the global host process remains. It can now read all local results $\mathbf{D}_i$ and $\boldsymbol{\sigma}_i$ from memory and continue to compute the global stiffness, eventually leading to the next step of the Newton iteration, the next load step and the final solution of the global problem.

8.4 Shared memory performance

In order to illustrate the superior performance of the shared memory implementation, when compared with a previous implementation using hard disk access, some benchmarks have been carried out. A shared memory benchmark program was implemented which creates and closes file mapping handles using `CreateFileMapping` and `MapViewOfFile`, and reads/writes data in the same way as the FEAP shared memory implementation does. Parameters can be set to control the size of data to read/write, the number of test runs and the number of employed OpenMP threads between which the data is distributed.

All tests have been run on a machine with an Intel i7-870 CPU [46], featuring 4 physical cores, 8 virtual cores using SMT and cache sizes of 32 KB L1 (data) and 256 KB L2 cache per core as well as an L3 cache of 8 MB shared among all cores. The theoretical maximum

memory bandwidth is 21 GB/s for this processor in dual channel mode, higher transfer rates can only be achieved when utilizing the various cache levels on the CPU instead of system memory. For comparison, an ordinary hard disk drive (HDD) reaches transfer rates of approximately 100 MB/s, or in the case of a modern solid state disk (SSD) up to 500-700 MB/s. The number of test runs for each data point in the benchmark was chosen sufficiently high to make sure that each individual test runs for at least one minute, in order to mitigate fluctuation effects and latency issues when measuring time.

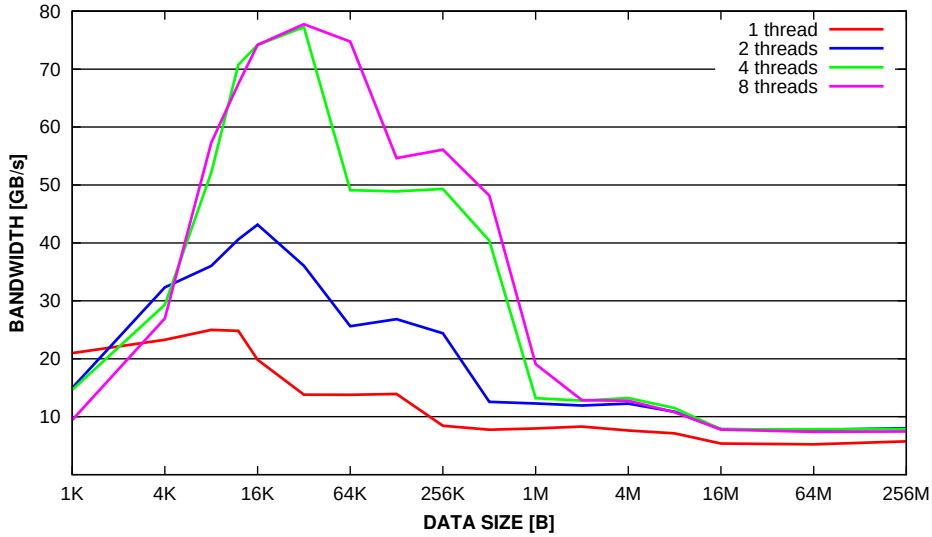


Figure 8.3: Shared memory bandwidth benchmark for varying data size and number of threads.

As seen in Figure 8.3, benchmarks for the implemented shared memory approach using the Windows API lead to real-world transfer rates between 5 and 80 GB/s, depending on parameters such as the size of written data (smaller data tends to be faster, as it fits into smaller cache levels) and the number of threads being used for the benchmark.

For very small data (< 8 KB), full performance can not yet be reached, especially when using multiple threads. The performance peaks somewhere between 12 – 32 KB data size, when all data is still small enough to fit in the L1 cache (32 KB per core). As the data is split and distributed to all available CPU cores, the performance drop when going from L1 to L2 cache occurs earlier when using less threads. For instance, when using a data size of 32 KB with 4 threads (equivalent to 4 CPU cores), only 8 KB are assigned to each core, which can still fit into L1. When only using 1 thread, however, the full data array with a size of 32 KB does not fit into L1 cache, as some further variables, e.g. for

controlling the program flow, must be stored there as well. Thus, the most significant performance drop occurs when increasing data size to 16 KB for 1 thread, to 32 KB for 2 threads, to 64 KB for 4 threads, and to 128 KB for 8 threads.

The same situation arises again when moving from the faster L2 to the slower L3 cache. Here, the performance is lowered significantly when increasing to data sizes of 256 KB (1 thread), 512 KB (2 threads) and 1 MB (4-8 threads, both corresponding to the maximum of 4 used CPU cores). The last performance drop arises when data becomes so large that it does not fit into L3 cache anymore (16 MB) and the system's main memory is used instead. As the L3 cache is shared between all cores, this drop occurs at the same time regardless of the number of utilized threads. The achievable performance then stagnates slightly below 8 GB/s when using at least 2 threads. It is noted that using only a single thread does not lead to optimal performance except for extremely small data sizes. In general, it is advisable to utilize all available threads for data transfer.

The situation comparing best to a practical usage scenario within FEAP would be utilizing all CPU cores in parallel (as data can be written in parallel by each process assigned to a local model) and using large data, as especially the history data can easily accumulate to several hundred MB for each thread. The largest examples in Chapter 6 resulted in history data amounting to 15 GB, distributed to 8 processes, that had to be written and read in each iteration step.

For this case, a transfer rate of roughly 8 GB/s can be reached consistently, representing satisfactory 38% of the peak performance of 21 GB/s. Thus, the memory-based data access is at least 80 times as fast as hard disk-based access or at least 10 times faster when using a SSD. For more modern machines, the performance increase is expected to be even more noticeable, as system memory bandwidth is continuously increasing on new platforms, while the transfer rates of hard disk drives in particular are stagnating. Higher bandwidth might be achieved for simulations with smaller data, but in these cases calculation times are less relevant to begin with. These estimations agree with measurements within FEAP, where the shared memory data transfer was about 10 times as fast as a hard-disk data transfer in a practical usage scenario. It is noted, though, that the time intervals in this case ranged at 10^{-5} to 10^{-4} seconds, where no accurate time measurement is possible. This issue is mitigated in the benchmark scenario, where many tests are run subsequently to eliminate problems with measurement accuracy or fluctuating performance.

As a conclusion, while the implemented shared memory model successfully reduces data transfer times by at least an order of magnitude and thus fulfills its purpose, the true performance bottlenecks within the coupled simulation presented in this work lie elsewhere.

From the perspective of a full coupled calculation, no significant reduction in calculation time is achieved. Further more recent approaches have been started and partially completed successfully within the research group, in order to improve the performance of the coupled model. Most significantly, avoiding the local mesh initialization and assembly process when starting FEAP, implemented by reutilization of arrays that are only computed once for a single RVE, turned out to lead to a noticeable performance increase.

Chapter 9

Conclusion and future perspectives

A coupled two-scale shell model was presented in this work, utilizing quadrilateral shell elements on both the global and the local scale. On the global scale, a sandwich shell structure is modeled in a quasi-homogenized sense, where the homogenization parameters are re-evaluated in every iteration step of a nonlinear simulation. The RVE on the local scale is modeled appropriately to reflect the sandwich microstructure, using shell elements for the face layers and the cell walls. Due to the coupled formulation, the global-local model is able to describe nonlinear system behavior not only on the global scale, such as global buckling, but also on the local scale for e.g. local facesheet wrinkling or occurrence of plastic flow.

Various examples have been shown in Chapter 6 for different types of microstructures. Specifically, one- and two-directionally stiffened structures and hexagonal honeycombs have been investigated in this work, but the proposed model is suited to model all kinds of stiffener and cell geometries or corrugated core sandwiches. Some of these can be generated by starting with the honeycomb cell introduced in Section 6.4 and just relocating the position of the point S where the three legs of the tripod unit cell intersect each other.

For each of the discussed examples, first an RVE was isolated and its material tangent matrix was investigated. Regardless of the choice of an admissible RVE, the matrix must yield the same components. Some of the entries could be validated according to analytical expressions for the uni- and bidirectionally stiffened sandwich structures, leading to exact or very accurate results. In these cases and with $\nu = 0$, $\mathbf{D}$ has only components on its diagonal or an additional component D_{45}. Due to the choice of boundary conditions, the shear stiffness components can not be reflected accurately. This effect is mitigated by the introduction of a shear correction factor. In the more complicated example of a hexagonal cell microstructure, no simple analytical results were available. Moreover, $\mathbf{D}$ is now fully populated, even if some coupling entries between membrane and bending/shear

modes are many orders of magnitude smaller than the non-coupling entries. It can be seen that starting with a sufficiently large RVE with at least 2×2 tripod unit cells, the relevant components of the matrix of linearized stress resultants remain constant.

After concluding the investigation of $\mathbf{D}$ for each of the example RVEs, a coupled global-local system is introduced. In the case of the unidirectionally stiffened sandwich, which can be regarded as a parallel arrangement of multiple I-profiles, a beam-like structure is considered in a four-point bending setup. For the bidirectionally stiffened and the hexagonal honeycomb sandwiches, a plate under constant area load is modeled with Navier-support on the outer edges. Symmetry along the x- and y-axes is exploited for the bidirectionally stiffened structure, but this symmetry is lost for the hexagonal setup.

As a first step for each type of microstructure, a model with linear geometry and an isotropic linear-elastic material law is simulated. The results of the coupled model are compared to large scale classical shell models and show very good agreement for the global deformation behavior of all examples. While it was not possible to compare local stress states between the two models in an exact way, comparisons between two points geometrically close to each other matched reasonably well. The number of equations could be reduced considerably in all cases, and additionally all local RVE systems are decoupled of each other, further reducing the solution time.

Various types of nonlinearities were investigated as a next step. For the unidirectionally stiffened sandwich beam, nonlinear geometry was introduced on the global scale, leading to the expected stiffening of the system due to membrane stresses. In this situation, the homogenized parameters still remain constant and thus allow to evaluate only one local system once.

This is not the case for the bidirectionally stiffened sandwich plate, where nonlinear geometry was introduced on the local scale along with a geometric imperfection. Here, the nonlinear effects lead to stability issues and the occurrence of local face layer buckling. Naturally, the stress resultants and linearized stress resultants depend on the local deformation state and must be evaluated individually for each integration point in each iteration step.

As a next example, physical nonlinearity in terms of an elasto-plastic material law is embedded in the bidirectionally stiffened RVE. The coupled model is able to accurately describe the resulting deformation behavior even for large deformations, as long as the plastic effects do not localize on the system boundaries. Three different examples are considered for this situation. In addition, the local stress state agrees reasonably well even for a strongly deformed structure with mostly plastic deformations.

It must be noted that the implementation of the coupled method in its current form within FEAP still includes some avoidable performance bottlenecks and is only in some cases superior to a direct solution of a full scale system. Performance improvements were carried out step-by-step in earlier works [83], in this work, and will be further carried out in future works. The proposed algorithm can be understood as a proof of concept, showing that very accurate results for nonlinear systems can be achieved with vastly reduced numerical calculation effort. The details for a maximum performance implementation remain to be developed in other works or in a commercial toolkit, as a purely performance-oriented implementation would be better suited to frameworks other than FEAP. However, even despite the not yet optimal performance for the solution of coupled models, the proposed method is very advantageous in that it represents a functional and fully automated parallel framework for generating coupled nonlinear multiscale solutions, without having to manually interact with the calculation or its results during runtime. Moreover, the possibility to pause a simulation and zoom in to the local state in an RVE corresponding to a chosen integration point is very comfortable, as it allows to visualize local effects and deformations.

In addition to deriving the theoretical fundament and introducing the coupled global-local method, some usage aspects have been highlighted: The script-based generation of meshes, as well as some concepts of parallel programming for the application of a finite element program, along with a description and benchmark of the shared memory data-transfer interface developed for this work.

One aspect of the global-local method that might need further research is the formulation of boundary conditions. Even when using the crosswise link conditions as presented in Chapter 5, the shear deformation modes are slightly inhibited within the RVE, leading in some cases to length-dependent shear stiffness components in the matrix of linearized stress resultants, which is then compensated with a suitable shear correction factor. Further investigating the choice of boundary conditions and possible alternatives remains an open field for research.

Moreover, it would evidently be possible to investigate various other examples of sandwich panels in application with the coupled method. On the basis of the mesh generation functionality provided for the quadrilateral and hexagonal sandwich cores, different core geometries can be modeled. It is possible to create degenerated hexagonal cells forming angled rectangles or non-convex hexagons just by relocating the intersection point S where the cell walls meet. Modifying the mesh generation scripts allows to model any other kind of sandwich core as well, such as tubular or corrugated cores. The latter would lend themselves to investigation as the core face sheet constraint is already incorporated

in the existing model. Further modifications could include varying the thickness of cell walls (which is commonly found in real-world sandwich panels due to their manufacturing technique), filling cells with a foam, using composites as materials for the face layers, or modeling the adhesive layer connecting face layers and sandwich core, which was assumed to be perfect in this work. Further validating the method with respect to experimental results would be desirable, given the evident difficulties of generating finite element reference solutions for fine cell structures.

Another possible improvement would be researching adaptive methods, in which the full FE2 coupling is only executed in heavily loaded or deformed parts of a structure. In other regions that are only subjected to small shell strains $\boldsymbol{\varepsilon}_i$, the material tangent solution $\mathbf{D}$ stemming from a linear RVE could be employed to update local stress resultants according to $\boldsymbol{\sigma}_i = \mathbf{D}\,\boldsymbol{\varepsilon}_i$. Only where a certain adaptivity criterion is fulfilled, the coupled local solution $\mathbf{D}_i$ for a specific integration point i would be generated. It is not trivial, however, to find a suitable adaptivity criterion and to ensure that convergence is maintained as soon as a specific RVE switches from the linearized approach towards the local coupled simulation. Some approaches conducted within the scope of this work, based on a strain-energy criterion, did not yet lead to a stable simulation.

Bibliography

[1] *Abaqus 6.14 Documentation*. Dassault Systèmes. 2014.

[2] H.G. Allen. *Analysis and Design of Structural Sandwich Panels*. Pergamon Press, 1969.

[3] H. Altenbach. *Kontinuumsmechanik*. Springer, 2012.

[4] K.-J. Bathe. *Finite-Elemente-Methoden*. Springer, 2002.

[5] E. Becker and W. Bürger. *Kontinuumsmechanik*. B. G. Teubner, 1975.

[6] T. Belytschko and I. Leviathan. Physical stabilization of the 4-node shell element with one point quadrature. *Computer Methods in Applied Mechanics and Engineering* 113 (3–4), pp. 321–350, 1994.

[7] T.N. Bitzer. *Honeycomb Technology: Materials, Design, Manufacturing, Applications and Testing*. Chapman & Hall, 1997.

[8] S. Borkar and A.A. Chien. The future of microprocessors. *Communications of the ACM* 54 (5), pp. 67–77, 2011.

[9] N. Büchter and E. Ramm. Shell theory versus degeneration - a comparison in large rotation finite element analysis. *International Journal for Numerical Methods in Engineering* 34 (1), pp. 39–59, 1992.

[10] C.R. Calladine. *Theory of Shell Structures*. Cambridge University Press, 2007.

[11] E. Carrera and S. Brischetto. A Survey With Numerical Assessment of Classical and Refined Theories for the Analysis of Sandwich Plates. *Applied Mechanics Reviews* 62 (1) 2008.

[12] A. Catapano and M. Montemurro. A multi-scale approach for the optimum design of sandwich plates with honeycomb core. Part I: homogenisation of core properties. *Composite Structures* 118, pp. 664–676, 2014.

[13] A. Catapano and M. Montemurro. A multi-scale approach for the optimum design of sandwich plates with honeycomb core. Part II: the optimisation strategy. *Composite Structures* 118, pp. 677–690, 2014.

[14] P. Chadwick. *Continuum Mechanics: Concise Theory and Problems*. Dover Publications, 1998.

[15] E.W.C. Coenen, V.G. Kouznetsova, and M.G.D. Geers. A multi-scale computational strategy for structured thin sheets. *International Journal of Material Forming* 1 (1), pp. 61–64, 2008.

[16] E.W.C. Coenen, V.G. Kouznetsova, and M.G.D. Geers. Computational homogenization for heterogeneous thin sheets. *International Journal for Numerical Methods in Engineering* 83 (8-9), pp. 1180–1205, 2010.

[17] E.N. Dvorkin and K-J. Bathe. A continuum mechanics based four-node shell element for general non-linear analysis. *Engineering computations* 1, pp. 77–88, 1984.

[18] F. Feyel and J.L. Chaboche. FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fiber SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering* 183, pp. 309–330, 2000.

[19] M.J. Flynn. Some computer organizations and their effectiveness. *IEEE Transactions on Computers* 21, pp. 948–960, 1972.

[20] K. Garikipati and T.J.R. Hughes. A variational multiscale approach to strain localization - formulation for multidimensional problems. *Computer Methods in Applied Mechanics and Engineering* 188 (1), pp. 39–60, 2000.

[21] M.G.D. Geers, E.W.C. Coenen, and V.G. Kouznetsova. Multi-scale computational homogenization of structured thin sheets. *Modelling and Simulation in Materials Science and Engineering* 15 (4), pp. 393–404, 2007.

[22] S. Ghosh, K. Lee, and S. Moorthy. Two scale analysis of heterogeneous elastic-plastic materials with asymptotic homogenization and Voronoi cell finite element model. *Computer Methods in Applied Mechanics and Engineering* 132 (1–2), pp. 63–116, 1996.

[23] S. Ghosh, K. Lee, and P. Raghavan. A multi-level computational model for multi-scale damage analysis in composite and porous materials. *International Journal of Solids and Structures* 38, pp. 2335–2385, 2001.

[24] L.J. Gibson and M.F. Ashby. *Cellular Solids: Structure and Properties*. Cambridge University Press, 2001.

[25] F. Gruttmann, R. Sauer, and W. Wagner. Theory and numerics of three-dimensional beams with elastoplastic material behavior. *International Journal for Numerical Methods in Engineering* 48, pp. 1675–1702, 2000.

[26] F. Gruttmann and W. Wagner. Structural analysis of composite laminates using a mixed hybrid shell element. *Computational Mechanics* 37, pp. 479–497, 2006.

[27] F. Gruttmann and W. Wagner. A coupled two-scale shell model with applications to layered structures. *International Journal for Numerical Methods in Engineering* 94, pp. 1233–1254, 2013.

[28] C. Helfen and S. Diebels. Numerical multiscale modelling of sandwich plates. *Technische Mechanik* 32 (2-5), pp. 251–264, 2012.

[29] C. Helfen and S. Diebels. A numerical homogenisation method for sandwich plates based on a plate theory with thickness change. *Zeitschrift für Angewandte Mathematik und Mechanik* 93 (2-3), pp. 113 –125, 2013.

[30] D. Heller. *inp2feap Github page*. 2015. URL: `https://github.com/dheller1/inp2feap`.

[31] D. Heller and F. Gruttmann. A coupled two-scale shell model for comb-like sandwich structures. *Proceedings in Applied Mathematics and Mechanics* submitted 2015.

[32] R. Hill. Elastic properties of reinforced solids: some theoretical principles. *Journal of the Mechanics and Physics of Solids* 11, pp. 357–372, 1963.

[33] G.A. Hoffmann. Poisson's ratio for honeycomb sandwich cores. *Journal of the Aerospace Sciences* 25 (8), pp. 534–535, 1958.

[34] J. Hohe. A direct homogenisation approach for determination of the stiffness matrix for microheterogeneous plates with application to sandwich panels. *Composites: Part B* 34, pp. 615–626, 2003.

[35] J. Hohe. *Elastizitätsverhalten von Sandwich-Zellkernen und zweidimensionalen Modellschäumen*. Shaker, 2003.

[36] J. Hohe and W. Becker. Effective elastic properties of triangular grid structures. *Composite Structures* 45, pp. 131–145, 1999.

[37] J. Hohe and W. Becker. Effective stress-strain relations for two-dimensional sandwich cores: Homogenization, material models, and properties. *Applied Mechanics Reviews* 55 (1), pp. 61–87, 2002.

[38] J. Hohe, C. Beschorner, and W. Becker. Effective elastic properties of hexagonal and quadrilateral grid structures. *Composite Structures* 46, pp. 73–89, 1999.

[39] J. Hohe and L. Librescu. Advances in the Structural Modeling of Elastic Sandwich Panels. *Mechanics of Advanced Materials and Structures* 11, pp. 395–424, 2004.

[40] G.A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. John Wiley and Sons, 2000.

[41] G. Horvay. Bending of honeycombs and of perforated plates. *ASME Journal of Applied Mechanics* 19, pp. 122–123, 1952.

[42] H.C. Hu. On some variational principles in the theory of elasticity and the theory of plasticity. *Scientia Sinica* 4, pp. 33–54, 1955.

[43] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering, 2000.

[44] T.J.R. Hughes, G.R. Feijoo, L. Mazzei, and J.-B. Quincy. The variational multiscale method - a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering* 166 (1), pp. 3–24, 1998.

[45] Intel Corporation. *Intel Pentium D Processor 800 Sequence Datasheet*. 2006. URL: `http://www.intel.com/Assets/PDF/datasheet/307506.pdf`.

[46] Intel Corporation. *Intel Core i7-800 and i5-700 Desktop Processor Series - Datasheet Volume 1*. 2010. URL: `http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/core-i7-800-i5-700-desktop-datasheet-vol-1.pdf`.

[47] Intel Corporation. *Intel Inspector Documentation*. 2015. URL: `https://software.intel.com/intel-inspector-xe/documentation/windows`.

[48] Intel Corporation. *Intel Xeon Phi Coprocessor x100 Product Family Datasheet*. 2015. URL: `https://www-ssl.intel.com/content/www/us/en/processors/xeon/xeon-phi-coprocessor-datasheet.html`.

[49] A.L. Kalamkarov. On the determination of the effective characteristics of cellular plates and shells of periodic structure. *Mechanics of Solids* 22, pp. 175–179, 1987.

[50] A.L. Kalamkarov and A.G. Kolpakov. *Analysis, Design and Optimization of Composite Structures*. John Wiley and Sons, 1997.

[51] Karlsruhe Institute of Technology, Institut für Baustatik. *FEAP - A Finite Element Analysis Program*. Feb. 2010.

[52] S. Kelsey, R.A. Gellatly, and B.W. Clark. The Shear Modulus of Foil Honeycomb Cores. *Aircraft Engineering and Aerospace Technology* 30 (10), pp. 294–302, 1958.

[53] R. Kienzler, H. Altenbach, and I. Ott. *Theories of Plates and Shells: Critical Review and New Applications*. Lecture Notes in Applied and Computational Mechanics. Springer, 2004.

[54] S. Klinkel, F. Gruttmann, and W. Wagner. A mixed shell formulation accounting for thickness strains and finite strain 3d material models. *International Journal for Numerical Methods in Engineering* 74, pp. 945–970, 2008.

[55] V.G. Kouznetsova, M.G.D. Geers, and W.A.M. Brekelmans. Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme. *International Journal for Numerical Methods in Engineering* 54 (8), pp. 1235–1260, 2002.

[56] V.G. Kouznetsova, M.G.D. Geers, and W.A.M. Brekelmans. Multi-scale second-order computational homogenization of multi-phase materials: a nested finite element solution strategy. *Computer Methods in Applied Mechanics and Engineering* 193 (48–51), pp. 5525–5550, 2004.

[57] A. Kuzmin, M. Luisier, and O. Schenk. Fast Methods for Computing Selected Elements of the Greens Function in Massively Parallel Nanoelectronic Device Simulations. *Euro-Par 2013 Parallel Processing*. Vol. 8097. Lecture Notes in Computer Science. Springer, 2013, pp. 533–544.

[58] L. Librescu and T. Hause. Recent developments in the modeling and behavior of advanced sandwich constructions: a survey. *Composite Structures* 48, pp. 1–17, 2000.

[59] J. Lubliner. *Plasticity Theory*. Dover Publications, 2008.

[60] L.E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice Hall, 1977.

[61] C.C. Mei and B. Vernescu. *Homogenization Methods for Multiscale Mechanics*. World Scientific, 2009.

[62] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard*. Version 3.1. 2015. URL: `http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf`.

[63] J.C. Michel, H. Moulinec, and P. Suquet. Effective properties of composite materials with periodic microstructure: a computational approach. *Computer Methods in Applied Mechanics and Engineering* 172, pp. 109–143, 1999.

[64] Microsoft Corporation. *CreateFileMapping function*. May 2015. URL: `https://msdn.microsoft.com/en-us/library/windows/desktop/aa366537.aspx`.

[65] Microsoft Corporation. *MapViewOfFile function*. May 2015. URL: `https://msdn.microsoft.com/en-us/library/windows/desktop/aa366761.aspx`.

[66] C. Miehe, J. Schröder, and J. Schotte. Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering* 171, pp. 387–418, 1999.

[67] R. von Mises. Mechanik der plastischen Formänderung von Kristallen. *Zeitschrift für angewandte Mathematik und Mechanik* 8 (3), pp. 161–185, 1928.

[68] M. Montemurro, A. Vincenti, and P. Vannucci. The Automatic Dynamic Penalisation method (ADP) for handling constraints with genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 256, pp. 70–87, 2014.

[69] G.E. Moore. Cramming more components onto integrated circuits. *Electronics* 38 (8), pp. 114–117, 1965.

[70] A.K. Noor, W.S. Burton, and C.W. Bert. Computational models for sandwich panels and shells. *Applied Mechanics Reviews* 49, pp. 155–199, 1996.

[71] C. Norris and W. Kommers. *Short-column compressive strength of sandwich constructions as affected by the size of the cells of honeycomb-core materials*. Tech. rep. 1817. Forest Products Laboratory Madison, Wisconsin, 1956.

[72] NVIDIA Corporation. *Tesla K40 GPU Active Accelerator Board Specification*. 2013. URL: `http://www.nvidia.com/content/PDF/kepler/Tesla-K40-Active-Board-Spec-BD-06949-001_v03.pdf`.

[73] OpenMP Architecture Review Board. *OpenMP Application Program Interface*. Version 4.0. 2013. URL: `http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf`.

[74] D.H. Pahr and F.G. Rammerstorfer. Buckling of Honeycomb Sandwiches: Periodic Finite Element Considerations. *Computer Modeling in Engineering and Sciences* 12 (3), pp. 229–241, 2006.

[75] P. Perzyna. Fundamental Problems in Viscoplasticity. *Advances in Applied Mechanics* 9, pp. 243–377, 1966.

[76] F.J. Plantema. *Sandwich Construction*. John Wiley and Sons, 1966.

[77] F.J. Pollack. New microarchitecture challenges in the coming generations of CMOS process technologies. *Proceedings of the 32nd annual ACM/IEEE international symposium on microarchitecture*. 1999.

[78] P. Ponte Castaneda. New variational principles in plasticity and their application to composite materials. *Journal of the Mechanics and Physics of Solids* 40 (8), pp. 1757–1788, 1992.

[79] Python Software Foundation. *The Python Language Reference*. May 2015. URL: `https://docs.python.org/2/reference/index.html`.

[80] E. Riks. The Application of Newton's Method to the Problem of Elastic Stability. *ASME Journal of Applied Mechanics* 39 (4), pp. 1060–1065, 1972.

[81] O. Schenk, M. Bollhöfer, and R.A. Römer. On Large-Scale Diagonalization Techniques for the Anderson Model of Localization. *SIAM Rev.* 50 (1), pp. 91–112, 2008.

[82] O. Schenk, A. Wächter, and M. Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Computational Optimization and Applications* 36 (2-3), pp. 321–341, 2007.

[83] M. Schürg. *Theoretical modeling and parallel programming of a nonlinear composite finite shell element based on a mixed global-local variational principle*. Vol. 27. Forschungsberichte des Instituts für Mechanik der Technischen Universität Darmstadt, 2012.

[84] J. Schütt. *Ein inelastisches 3D-Versagensmodell für Beton und seine Finite-Element-Implementierung*. Vol. 9. Forschungsberichte des Instituts für Baustatik, Universität Karlsruhe (TH), 2005.

[85] H.-G. Sehlhorst, R. Jänicke, A. Düster, E. Rank, H. Steeb, and S. Diebels. Numerical investigations of foam-like materials by nested high-order finite element methods. *Computational Mechanics* 45 (1), pp. 45–59, 2009.

[86] R.L. Sierakowski and J.R. Vinson. *The Behavior of Structures Composed of Composite Materials*. Springer, 2002.

[87] Silverfrost Limited. *FTN95 Fortran compiler*. Version 6.10.0. 2011. URL: `http://www.silverfrost.com/ftn95-help/ftn95_contents.aspx`.

[88] J.C. Simo. On a stress resultant geometrically exact shell model. Part VII: shell intersections with 5/6-DOF finite element formulations. *Computer Methods in Applied Mechanics and Engineering* 108, pp. 319–339, 1993.

[89] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Springer, 2000.

[90] I.C. Skrna-Jakl, D.H. Pahr, K.H. Karner, and F.G. Rammerstorfer. Homogenization Technique for the Efficient Modeling of Large Scale Corrugated Core Sandwich Structures. *International Journal of Applied Physics and Mathematics* 4 (1), pp. 62–67, 2014.

[91] K. Terada and N. Kikuchi. A class of general algorithms for multi-scale analysis of heterogeneous media. *Computer Methods in Applied Mechanics and Engineering* 190, pp. 5427–5464, 2001.

[92] The MathWorks Inc. *MATLAB Version R2013a*. 2013. URL: `http://www.mathworks.com/help/matlab/index.html`.

[93] S. Timoshenko. *Theory of Elasticity*. McGraw-Hill, 1970.

[94] W. Wagner and F. Gruttmann. A robust non-linear mixed hybrid quadrilateral shell element. *International Journal for Numerical Methods in Engineering* 64, pp. 635–666, 2005.

[95] K. Washizu. *On the variational principles of elasticity and plasticity*. Tech. rep. 25–18. Aeroelastic and Structures Research Laboratory, MIT, Cambridge, Massachusetts, 1955.

[96] W.K. Wonach and F.G. Rammerstorfer. A general approach to the wrinkling instability of sandwich plates. *Structural Engineering and Mechanics* 12, pp. 363–376, 2001.

[97] P. Wriggers. *Nonlinear Finite Element Methods*. Springer, 2008.

[98] H. Yang, S. Saigal, A. Masud, and R.K. Kapania. A survey of recent shell finite elements. *International Journal for Numerical Methods in Engineering* 47 (1–3), pp. 101–127, 2000.

[99] S. Yusuff. Theory of wrinkling in sandwich construction. *Journal of the Royal Aeronautical Society* 59, pp. 30–36, 1955.

[100] D. Zenkert. *Handbook of Sandwich Construction*. Engineering Materials Advisory Services, 2008.

[101] J. Zhang and M.F. Ashby. The out-of-plane properties of honeycombs. *International Journal of Mechanical Sciences* 34, pp. 475–489, 1992.

[102] O.C. Zienkiewicz, R.L. Taylor, and D.F. Fox. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier Butterworth-Heinemann, 2013.

[103] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, 2013.

Bisher sind in dieser Reihe erschienen

Band 1
Zur mikrorissinduzierten Schädigung spröder Materialien
B. Lauterbach, Dissertation 2001, ISBN 3-935868-01-4

Band 2
3D-Simulation der Mikrostrukturentwicklung in Zwei-Phasen-Materialien
R. Müller, Dissertation 2001, ISBN 3-935868-02-2

Band 3
Zur numerischen Simulation von Morphologieänderungen in mikroheterogenen Materialien
S. Kolling, Dissertation 2001, ISBN 3-935868-03-0

Band 4
Theoretische und numerische Untersuchung von Versagensmechanismen in Metall-Keramik-Verbundwerkstoffen
T. Emmel, Dissertation 2002, ISBN 3-935868-04-9

Band 5
On microcrack dominated problems in dynamics and statics of brittle fracture: a numerical study by boundary element techniques
S. Rafiee, Dissertation 2002, ISBN 3-935868-05-7

Band 6
Kontinuumsmechanik anisotroper Festkörper und Fluide
H. Ehrentraut, Habilitationsschrift 2002, ISBN 3-935868-06-5

Band 7
Plane unsteady inviscid incompressible hydrodynamics of a thin elastic profile
N. Blinkova, Dissertation 2002, ISBN 3-935868-07-3

Band 8
Anmerkungen zur Simulation von entfestigendem Materialverhalten
H. Baaser, Habilitationsschrift 2004, ISBN 3-935868-08-1

Band 9
Orts- und zeitadaptive DAE-Methoden zur Beschreibung elastisch-plastischen Materialverhaltens innerhalb der FEM
S. Eckert, Dissertation 2005, ISBN 3-935868-09-X

Band 10
Simulations of the Flow of the Ross Ice Shelf, Antarctica: Parameter Sensitivity Tests and Temperature-Dependent Rate Factor
A. Humbert, Dissertation 2005, ISBN 3-935868-10-3

Band 11
A Thermo-mechanical Continuum Theory with Internal Length of Cohesionless Granular Materials
Chung Fang, Dissertation 2006, ISBN 3-935868-11-1

Band 12
Modeling Dry Granular Avalanches past Different Obstructions: Numerical Simulation and Laboratory Analyses
Chiou Min-Ching, Dissertation 2006, ISBN 3-935868-12-X

Band 13
Configurational forces in defect mechanics and in computational methods
R. Müller, Habilitationsschrift 2005, ISBN 3-935868-13-8

Band 14
Hyperelastic dynamics in physical and material space
S. Kolling, Habilitationsschrift 2007, ISBN 978-3-935868-14-3

Band 15
Phenomenological modeling of ferroelectric material behavior
V. Mehling, Dissertation 2007, ISBN 978-3-935868-15-0

Band 16
Ein mischungsbasiertes Materialmodell zum Knochenumbau
R.-R. Kühn, Dissertation 2006, ISBN 978-3-935868-16-7

Band 17
Einige Erweiterungen der Rand-Finite-Elemente-Methode und deren Anwendung auf Randeffekte in ebenen Laminaten
J. Artel, Dissertation 2007, ISBN 978-3-935868-17-4

Band 18
Spannungskonzentrations-Effekte an Verstärkungspflaster-Ecken
H. Wigger, Dissertation 2008, ISBN 978-3-935868-18-1

Band 19
Rotationseffekte in der Kristallplastizität
C. Bröse, Dissertation 2007, ISBN 978-3-935868-19-8

Band 20
Finite-Element-Modelle zur Simulation von Delaminationen dünner Filme auf Substraten
V. D. Pham, Dissertation 2010, ISBN 978-3-935868-20-4

Band 21
Asymptotische Nahfeldanalysen ebener Multi-Materialverbindungsstellen mit der Methode komplexer Potentiale
C. Sator, Dissertation 2010, ISBN 978-3-935868-21-1

Band 22
Modellierung spröder Rissbildung an Spannungskonzentrationen mit der Bruchmechanik finiter Risse
J. Hebel, Dissertation 2010, ISBN 978-3-935868-22-8

Band 23
Some Contributions to the Homogenization of Macroscopically Isotropic Composites
V. Salit, Dissertation 2011, ISBN 978-3-935868-23-5

Band 24
Asymptotic Analysis of the Load Transfer on Double-Lap Bolted Joints
J. Kratochvíl, Dissertation 2012, ISBN 978-3-935868-24-2

Band 25
Spannungssingularitätsordnungen in linear-elastischen und piezoelektrischen Multimaterialkonfigurationen mit der Rand-Finite-Elemente-Methode
W. Mayland, Dissertation 2012, ISBN 978-3-935868-25-9

Band 26
Plastizität und Skaleneffekte sowie Deformations- und Versagensmodellierung dünner metallischer Schichten bei Nanoindentation
A. Trondl, Dissertation 2012, ISBN 978-3-935868-26-6

Band 27
Theoretical modeling and parallel programming of a nonlinear composite finite shell element based on a mixed global-local variational principle
M. Schürg, Dissertation 2012, ISBN 978-3-935868-27-3

Band 28
Strukturmechanische Modellierung und Analyse des Tragverhaltens von dünnwandigen hochbelasteten Composite-Biege- und Querkraftträgern
A. M. Kroker, Dissertation 2013, ISBN 978-3-935868-28-0

Band 30
Der Laminatrandeffekt und seine Analyse, insbesondere mit der Rand-Finite-Elemente-Methode
J. Lindemann, Dissertation 2013, ISBN 978-3-935868-30-3

Band 31
Avoidance of brake squeal by a separation of the brake disc's eigenfrequencies: A structural optimization problem
A. Wagner, Dissertation 2013, ISBN 978-3-935868-31-0

Band 32
Ultrasonic Generators for Energy Harvesting Applications: Self-Excitation and Mechanical Frequency Transformation
E. Heffel, Dissertation 2013, ISBN 978-3-935868-32-7

Band 33
Neue Ansätze zur Analyse der Lastübertragung und Initiierung finiter Risse in Klebverbindungen
P. Weißgraeber, Dissertation 2014, ISBN 978-3-935868-33-4

Band 34
Instabilities and Wear Propagation in Calenders: Interactions with Structural Dynamics and Contact Kinematics
M. Eckstein, Dissertation 2014, ISBN 978-3-935868-34-1

Band 35
Adaptive Camber Airfoil for Load Alleviation in Horizontal Axis Wind Turbines: Analytical and Numerical Study
H. Spiegelberg, Dissertation 2014, ISBN 978-3-935868-35-8

Band 36
Erweiterungen der Rand-Finite-Elemente-Methode zur Analyse von Platten und Laminaten mit besonderem Fokus auf der Ermittlung von Singularitätsordnungen an Rissen und Kerben
R. Dieringer, Dissertation 2015, ISBN 978-3-935868-36-5

Band 37
Entwicklung und Analyse mikromechanischer Modelle zur Beschreibung des Effektivverhaltens von geschlossenzelligen Polymerschäumen
N.-C. Fahlbusch, Dissertation 2015, ISBN 978-3-935868-37-2

Band 38
Reduktion niederfrequenter Schwingungen von Windenergieanlagen durch Tilgersysteme
S. Katz, Dissertation 2015, ISBN 978-3-935868-38-9

Band 39
Multistable Structures for Broad Bandwidth Vibration-based Energy Harvesters: An Analytical Design Investigation
M. Heymanns, Dissertation 2015, ISBN 978-3-935868-39-6